PROBABILISTIC MODELS FOR MULTI-CLASSIFIER BIOMETRIC AUTHENTICATION USING QUALITY MEASURES

THÈSE N^{O} 3954 (2007)

PRÉSENTÉE LE 7 DÉCEMBRE 2007

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

Institut de traitement des signaux

SECTION DE GÉNIE ÉLECTRIQUE ET ÉLECTRONIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Jonas RICHIARDI

M.Phil., Darwin College, University of Cambridge, Royaume-Uni
de nationalité suisse et originaire de Genève (GE)

acceptée sur proposition du jury:

Prof. J. R. Mosig, président du jury
Dr A. Drygajlo, directeur de thèse
Prof. J. Kittler, rapporteur
Prof. J. Ortega-Garcia, rapporteur
Prof. J.-Ph. Thiran, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Lausanne, EPFL
2008

[...] rasend wär ich, das müsst ihr selbst gestehn, wenn ich im ganzen Gebiet der Möglichkeit mich nicht versuchte.

Heinrich Von Kleist, *Penthesilea*, scene IX

Contents

Acknowledgements

First, I wish to sincerly thank my thesis adviser Dr. Andrzej Drygajlo for his guidance, support, and confidence throughout my time at EPFL. His friendship went well beyond supervisor duties.

Second, my thesis committe, Prof. Kittler, Prof. Ortega-Garcia, and Prof. Thiran deserve my sincere thanks for taking the time to review this thesis and their constructive comments, and Profs. Kittler and Thiran deserve extra thanks for having sat through the "great ITS thesis deluge of October 2007".

I would like to heartfully thank my family in Geneva for their unfailing support since 1976. May the young ones and the slightly less so travel far, learn much, listen attentively, and above all, keep playing with the world around them.

To the Geneva Crew (Nicolas, Philippe, David, Carl, and their respective partners and children), I am grateful for knowing the difference between '89 and '90. To Cédric I am grateful for his mind-controlling powers.

To my dear friends in Lausanne, Serge, Lionel and Sladjana, Julien, and Sébastien, your presence in this city over the past years has made it seem (almost) as good as Geneva.

I have been very fortunate to be part of the Débiteurs theatre company over the past years, and I wish to extend my warmest thanks for the incredible *saltimbanque* adventures that were *La Thébaïde* and *Electre* to Jérome, Mathieu, Gaël, Julien and Olivia, as well as to our dear mother Margarita. Here is to many more plays, and to our next show in *La cour du palais des papes*.

My two friends and fitness trainers Patricia and Lorenzo P. deserve my special thanks for trying to keep the Ph.D.-induced body fat in check, and for making me overcome my childhood abhorrence of summer mountain sports (8b+ coming right up, just one more 6a please). In that respect, Giulia, Klaus, Carlo, Marco, Stefano and Karin deserve no thanks, as they have spent every opportunity trying to enlarge my waistline. The main culprits being of course Gaetano 'u zappatore' and Joanna, Gianluca 'fiorentina', Lorenzo G. 'saperlipopette' and Cristina.

The improvisational nebula of Lausanne, the PIP, LDS and Improsteurs teams have kept me in love with the stage, and I wish to thank them for sharing some of the best moments in my life, including our moral victory at Mondialito and the Royal Rumble, and the artery-clogging *mitraillettes de chez Billy*.

Jean-Paul, Jérome, and Jean, the three quarters of the J's of the EPFL breakdance crew, deserve a penguin slide for keeping it happening on and off the floor.

The colleagues and friends at the signal processing institute of EPFL have made work a pleasure, and I wish to thank my thesis twin KK, whose name I shall someday be able to write without hesitation. I am very much indebted to Anil Alexander and Plamen Prodanov for long blackboard debates over practical speech signal processing, statistics, modelling issues, life, and prosciutto. We have shared some memorable moments with other colleagues at the institute, and Julien (100% pozor), Ulrich, Mathieu, Yannick, Yann, Dan, Matteo, and others have contributed to a very friendly

atmosphere in and out of the lab. In another corner of the technical universe, I thank Martin and David for still being DTI at heart and enthusiastic about it.

Our technical and administrative staff has been very friendly too, and I thank Gilles Auric and Eric Gruaz for their technical help and discussions, Marianne for helping me navigate the administrative jungle at the beginning and Chantal for the same, but later.

I am very grateful to have met many intereseting and friendly researchers all across Europe along the way, with which I had many fruitful discussions: Jérome Louradour at IRIT (I shall remember that duck sausage for quite some time), Enrique Argones-Rua at the University of Vigo (my revenge at pool is on its way), Julian Fierrez-Aguilar and Daniel Ramos-Castro at the Autonomous University of Madrid, Harold Mouchère at IRISA, Dr. Jean Hennebert at the University of Fribourg, Nick Evans at the University of Wales Swansea, and Nicolas Scheffer at LIA-UAPV all contributed in shaping my understanding of pattern recognition.

Throughout my studies, I have also met a number of more senior researchers with whom I learned a great amount about engineering. I would like to thank Prof. Larry Lind at the University of Essex for first sparking my interest in signal processing, as I believe few people have such capacity to convey enthusiasm about radix-2 FFT to a class of undergraduates. Likewise, Prof. Ann Copestake at the university of Cambridge deserves commendation for making topics such as part-of-speech tagging thoroughly enjoyable. Dr. David Barber at IDIAP proposed a very rigorous and deeply satisfying approach to learning and inference in graphical models in his doctoral course. Dr. Sami Bengio has many times asked pointed questions that sent me back to the drawing board. It was also enriching to meet Prof. Ian Witten of the university of Waikato and Prof. Ethem Alpaydin of Bogazici University during the Biosecure summer workshop, and to witness their practical approach firsthand. Indeed, *try simple things first*.

I would also like to congratulate and extend many thanks to Prof. Kevin Murphy for having the marvellous idea of making the Bayes Net Toolbox available, without which this thesis would probably have taken much longer.

Last but absolutely not least, my heartfelt thanks to Pelin for bearing with me (and actually a fair bit more than just that) during the writing, and being such an ekmek in general.

Abstracts

English abstract

Biometric authentication can be cast as a signal processing and statistical pattern recognition problem. As such, it relies on models of signal representations that can be used to discriminate between classes. One of the assumptions typically made by the practioner is that the training set used to learn the parameters of the class-conditional likelihood functions is a representative sample of the unseen test set on which the system will be used. If the test set data is distorted, the assumption no longer holds and the Bayes decision rule or Maximum Likelihood rules are no longer optimal. In biometrics, the distortions of the data come from two main sources: intra-user variability, and changes in acquisition conditions. The aim of the thesis is to increase robustness of biometric verification systems to these sources of variability.

Since the signals under consideration are not deterministic, but stochastic, steady-state signal analysis techniques are not adequate for modelling. By using probabilistic methods instead, we can obtain models describing, amongst other, the amount of spread in the random variables, meaning that we can take into account the uncertainty on the realisation of the random variables (features) due to intra-user variability. Furthermore, we posit that modelling information reflecting the acquisition conditions (signal quality measures) should be useful in improving the robustness of biometric verification systems to changes of data from the training conditions.

In this thesis, we use probabilistic approaches at all stages of the biometric authentication processing chain, while taking into account the quality of the signal being modelled. We use the theoretical framework of Bayesian networks, a family of graphical models offering important flexibility. We use them both for single-classifier systems (base classifier and reliability model) and for multiple-classifier systems (classifier combination with and without quality measures).

In the single-classifier part, we propose to use a Bayesian network topology equivalent to a Gaussian mixture model for signature verification, and show that experimental results are equivalent to state-of-the-art signature verification systems. Furthermore, the model can be used for speaker verification as well.

Quality measures are auxiliary information that can be used in both single-classifier systems and multi-classifier systems. We define precisely the concept of quality measure, and show the different potential types of quality measures. We propose new quality measures for both speech and signature, as well as the concept of modality-independent quality measure, as an additional type of auxiliary information. We show that the effect of signal degradation could be different on impostor and client score distributions, an important effect to take into account when designing quality-based fusion models. We propose a principled evaluation methodology for quality measures.

The use of reliability models is proposed. They are probabilistic models of single-classifier be-

haviour, taking into account quality measures. They result in an enhanced confidence measure, which is to some degree robust with respect to changing quality. Experiments show that reliability estimation generally outperforms confidence estimation.

We formalise different classifier combination algorithms as probabilistic models in the framework of Bayesian networks for both decision-level and score-level fusion, and propose enhancements to existing models. We also propose a new structure learning algorithm, sparse regression fusion (SRF), specifically designed for classifier combination tasks. The SRF model obtains good results over three multimodal benchmark databases.

Lastly, we propose a theoretical view on probabilistic classifier combination with quality measure, based on an analysis of independence and conditional independence relationships induced by different model topologies. We also show the importance of the notion of context-specific independence, and draw a parallel between decision tree building and enforcing a weak version of context-specific independence. Three quality-based fusion schemes are proposed: SRF-Q, an adaptation of the SRF algorithm to the use of quality measures, Context-specific fusion with quality measures (CSF-Q), a fusion model equivalent to a decision tree but motivated by probabilistic and independence arguments, and rigged majority voting, a flexible scheme that can be used with both reliability models and other meta-classifiers, with clear limits on accuracy gains that can be expected. The CSF-Q and the SRF-Q algorithms perform better than state-of-the-art combiners not using quality measures, and under certain conditions better than existing state-of-the-art combiners using quality measures.

Keywords: multiple classifiers, probabilistic models, pattern recognition, quality measures, Bayesian networks, multimodal, biometrics, signature, speech

Version abrégée française

La vérification biométrique d'identité peut être vue comme un problème de traitement du signal et de reconnaissance des formes statistique. En tant que tel, elle se base sur des modèles de représentations de signaux qui peuvent être utiliser pour discriminer entre des classes. Un des présupposés généralement employé par le praticien est que l'ensemble de données d'entraînement utilisé pour apprendre les paramètres du modèle constitue un échantillon représentatif de l'ensemble de test caché sur lequel le système sera testé. Si l'ensemble de test est distordu, ce présupposé n'est plus applicable, et la règle de décision de Bayes, ou la règle de la vraisemblance maximale, ne sont plus optimales. En biométrie, les distortions dans les données proviennent de deux sources principales: la variabilité interne à l'utilisateur, et le changement dans les conditions d'acquisitions du signal. L'object de la présente thèse est d'améliorer la robustesse des systèmes de vérification biométriques à ces sources de variabilité.

Comme les signaux en cause ne sont pas déterministes, mais stochastiques, les techniques d'analyse de signal fixe ne sont pas applicable pour la modélisation. En utilisant des méthodes probabilistes, nous obtenons des modèles décrivant, entre autres, l'écart-type des variables aléatoires, ce qui signifie que l'on peut prendre en compte l'incertitude liée à la réalisation de la variable aléatoire (paramètre) due à la variabilité interne à l'utilisateur. De plus, nous supposons que la modélisation d'information reflétant les conditions d'acquisitions du signal pourraient être utiles pour améliorer la robustesse des systèmes de vérification d'identité biométrique aux changement de la distribution des données par rapport aux distributions d'entraînements.

Dans cette thèse, nous utilisons des approches probabilistes à toutes les étapes du processus de traitement biométrique, en prenant en compte la qualité du signal modélisé. Nous utilisons le cadre théorique des réseaux Baysésiens, un membre de la famille des modèle graphiques qui offre une souplesse importante. Nous utilisons les réseaux de Bayes aussi bien pour les systèmes à un seul classifieur (classifieur de base et modèle de fiabilité) que pour les sytèmes à classifieurs multiples (combinaison de classifieur avec et sans mesures de qualité).

Dans la partie traitant des systèmes à un seul classifieur, nous proposons l'utilisation d'une topologie de réseau de Bayes équivalente à un modèle à mélange de Gaussiennes, pour la vérification de signature, et nous montrons que les résultats expérimentaux sont équivalent aux résultats de pointe. De plus, le même modèle peut être utilisé pour la vérification du locuteur.

Les mesures de qualité sont une information auxiliaire qui peut être utilisée aussi bien dans les systèmes à un seul classifieur que dans les systèmes à classifieurs multiples. Nous définissons précisément le concept de mesure de qualité, et montrons les différents types potentiels de mesures de qualité. Nous proposons des nouvelles mesures de qualité pour la voix et la signature, et introduisons le concept de mesure de qualité indépendante de la modalité. Nous montrons que l'effet d'une dégradation du signal peut être différente sur les distributions des scores des clients et sur celles des imposteurs; ceci est un effet important à considérer lors de la conception de modèles de fusion basés sur la qualité. Nous proposons une méthodologies d'évaluation pour les mesures de qualité.

Nous proposons l'utilisation des modèles de fiabilité. Ce sont des modèles probabilistes du comportement de classifieurs de base, qui prennent en compte les mesures de qualité. Leur application résulte en des estimations de confiance améliorées, qui est quelque peu robuste aux changement de conditions d'acquisition. Les expériences montrent que l'estimation de la fiabilité donne généralement des meilleurs résultats que l'estimation de confiance.

Nous formalisons plusieurs algorithmes de combinaisons de classifieurs en tant que modèles probabilistes dans le cadre théorique des réseaux de Bayes, aussi bien pour la fusion au niveau des décisions que pour la fusion au niveau des scores. Nous proposons des améliorations à des modèles existants. Nous proposons également un nouvel algorithme d'apprentissage de structure, l'algorithme de fusion

par régression à densité faible (SRF), qui est conçu spécialement pour les tâches de combinaison de classifieurs. Cet algorithme obtient des bons résultats sur trois bases de données multimodales de référence.

Pour terminer, nous proposons un regard théorique sur la combinaison probabiliste de classifieurs avec des mesures de qualité, basée sur une analyse des relations d'indépendence et d'indépendence conditionnelle induite par différente topologies de modèle. Nous montrons également l'importance de la notion d'indépendence spécifique au contexte, et traçons un parallèle entre la construction d'arbres de décision et la mise en oeuvre d'une version faible de l'indépendence spécifique au contexte. Nous proposons trois modèles de fusion basée sur la qualité: Le modèle SRF-Q, qui est une adaptation de l'algorithme SRF pour l'utilisation des mesures de qualité. CSF-Q, un modèle de fusion équivalent à un arbre de décision, mais motivé par des arguments probabilistes et d'indépendence, et le modèle de vote majoritaire truqué, un modèle de fusion flexible qui peut s'utiliser soit avec des modèles de fiabilité, soit avec de méta-classifieurs, avec des limites claires sur les gains qui peuvent être attendus. Les modèles CSF-Q et SRF-Q donnent de meilleurs résultats que des combineurs de pointe qui n'utilisent pas de mesures de qualité, et sous certaines conditions de meilleurs résultats que les combineurs de pointe utilisant les mesures de qualité.

Mots-clé: classifieurs multiples, modèles probabilistes, reconnaissance de formes, mesures de qualité, réseaux de Bayes, multimodalité, biométrie, signature, parole

List of Figures

List of Tables

Notation

Uppercase variables generally represent random variables, while lowercase variables represent measurements on (or instantiations of) that variable.

CID	Classified IDentity (classifier decision)
D	dimensionality of a feature vector, number of attributes
DR	Decision Reliable (indicates $CID = \Omega$)
L	number of classifiers in a classifier ensemble
M	number of mixture components, mixture node
Ω, ω	class variable
$\mathbf{O}_t$	t^{th} feature vector in a sequence
$\phi(X)$	probability potential defined over the domain of random variable X
T	length of an observation sequence, number of data points, number of cases
τ	threshold (decision threshold or independence threshold)
Θ	model, set of parameters
TID	True IDentity (alternative notation for class variable Ω)
U	number of users
$A \perp\!\!\!\perp B \mid C$	A is independent from B given C
$A \not\perp\!\!\!\perp B$	A is not independent from B
A'	matrix or vector transpose of A
$\stackrel{\triangle}{=}$	is equal by definition, denotes

1 Introduction

Biometric verification is a fascinating and challenging problem to work on. It sits at the crossroads of many disciplines, both in hard sciences and social sciences. Signal processing and pattern recognition, the two essential elements of biometric verification, are themselves mixed engineering disciplines, built on mathematical tools such as probability theory, graph theory, information theory, and statistics. But biometric verification straddles and interacts with many other disciplines of research: the insights and techniques developed in the fields of human-computer interaction, ergonomics, and cryptography are now essential components in biometric systems. It is also at the core of many ethical and legal issues in the ever-increasingly digital world of today: here, as often, technology is leading the way and public debate is lagging, if not lacking entirely.

The field pioneered by Alphonse Bertillon (the son of a statistician) in the late 19th century has indeed grown enormously, benefiting from properly established scientific principles, and has become an important source of revenue for specialised companies. This apparent maturity has prompted the emergence of large-scale applications, such as biometric identity documents, for which more problems will have to be solved. In general, and while steady progress is registered each year, real-world deployments of biometric verification systems perform significantly worse than those tested in laboratory conditions.

In this Chapter, we start by defining essential terms in Section 1.1. We then cast biometric verification as a signal processing and pattern recognition task (Section 1.2). Section 1.3 introduces the problem setting for this thesis, leading to the definition of objectives in Section 1.4. Section 1.5 lists the major contributions of this thesis, and Section 1.6 gives an overview of the remainder of the thesis.

1.1 Biometrics and Identity

The term *identity* is defined as "the quality or condition of being the same in substance, composition, nature, properties, or in particular qualities under consideration" [288]. The *personal identity* is thus a data set that allows to recognize a person and to distinguish her from another one, and that can establish the identity of this person.

1.1.1 Identity proof

Three approaches are possible to prove a person's identity [194] and to provide "the right person with the right privileges the right access at the right time" [317]. These identity proving approaches, which establish the genuineness of the identity, are typically defined in colloquial terms as:

Something you have : The associated service or access is received through the presentation of a physical object (keys, smart card, identity document, ...), in *possession* of the concerned person.

Something you know : Pre-defined *knowledge*, such as a password normally kept secret, permits access to a service.

Something you are : Measurable personal traits, such as *biometric* characteristics, can also be used for identity proof.

A combination of these approaches makes the identity proof more secure. The use of the third approach, in addition to the others, has significant advantages. Without sophisticated means, biometrics are difficult to share, steal or forge and cannot be forgotten or lost. The development of methods for performing automatic biometric identity verification is the aim of this thesis.

1.1.2 Biometric identity verification

Biometry is a term whose first historical meaning was "the application of modern statistical methods to the measurements of biological objects" [288]. By language evolution, the term biometrics nowadays usually refers to automatic technologies for measuring and analyzing biological and anthropological characteristics such as fingerprints, irises, speech, face, and hand measurements, especially for identity proof. The current definition of biometrics refers to "[...] identifying an individual based on his or her distinguishing characteristics" [29].

The biometric identity verification task can be formally defined as follows: given a sequence of features $\mathbf{O}$ and a claimed user u, decide whether u indeed produced the sequence of features $\mathbf{O}$. Given a model $\mathbf{\Theta}^u$ for the claimed user and $\mathbf{\Theta}^-$ an antithetical model, a score function $S(\mathbf{O}, \mathbf{\Theta}^u, \mathbf{\Theta}^-)$, which for statistical models is generally a likelihood ratio, is used to determine whether the score of the input modality signal is above or below some threshold $\mathcal{T}$:

$$S(\mathbf{O}, \mathbf{\Theta}^u, \mathbf{\Theta}^-) \begin{cases} \geq \mathcal{T} & \text{accept identity claim} \\ < \mathcal{T} & \text{reject identity claim} \end{cases} \tag{1.1}$$

1.2 Biometrics as a signal processing and pattern recognition task

A *biometric verification system* is essentially a pattern classification system. As such, biometric verification follows the three phases of any pattern recognition system [136]: data acquisition and preprocessing, data representation, and decision-making. More precisely, for all modalities, biometric data will be transformed according to the processing steps described below and summarised in the activity diagram of Figure 1.1.

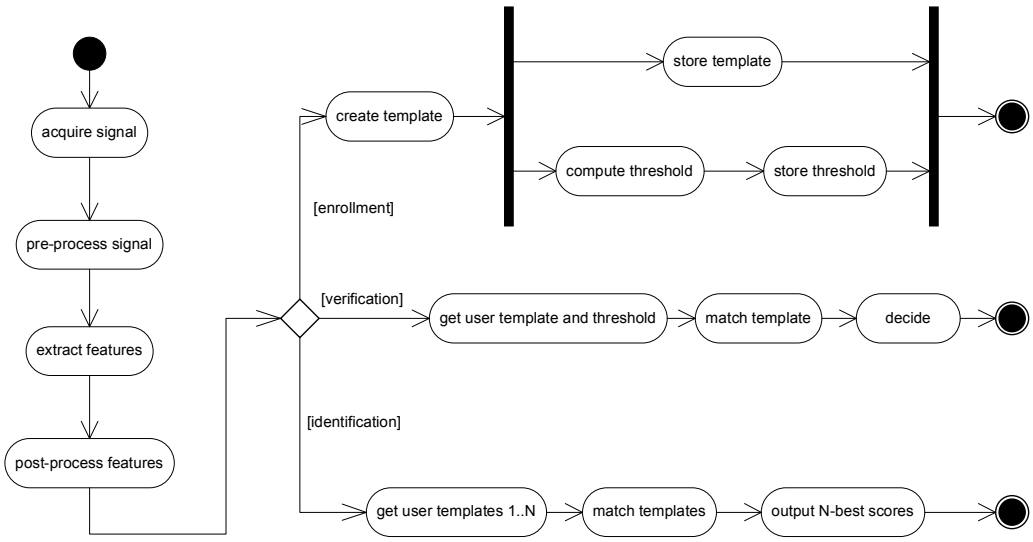


Figure 1.1 — UML activity diagram of processing steps for enrollment, verification, and identification in single-classifier biometric recognition systems.

1.2.1 Processing steps for single-classifier biometric pattern recognition

Capture or acquisition The biometric data (voice, on-line signature, fingerprint, ...), also called biometric presentation, are digitised via the input device (microphone, pen tablet, fingerprint scanner, ...) and stored in memory.

Preprocessing The signal-domain acquired data is prepared for feature extraction. This is typically used for normalising the signal-domain data and remove biases or sources of corruption in a systematic fashion. For speech, this includes for instance DC component removal as well as silence detection and removal. For signatures, this stage would include translating the signature to start at $(0, 0)$ coordinates and resampling the signature. For fingerprints, this may include rotation normalisation and thinning (skeletarisation).

Feature extraction Discriminative features are extracted from the preprocessed data. Although features are very different for each biometric modality, the general underlying principle remains the same: this processing step typically reduces the dimensionality of the input data to create a feature-level representation of input patterns that will be used by the classifier to perform pattern recognition. Typical examples of features include Mel Frequency Cepstral Coefficients or Perceptual Linear Prediction coefficients for speech, tangent angles and velocities for on-line signature, and minutiae locations for fingerprint: "In general, feature extraction is a form of non-reversible compression, meaning that the original biometric image cannot be reconstructed from the extracted features" [312].

Postprocessing Features are normalised to remove bias or adapt them to the classifier. An example of removing feature-domain bias is cepstral mean subtraction for speech, where transmission channel effects can be compensated for. Additionally, certain classifiers such as neural networks or support vector machines work best when their inputs have comparable dynamic ranges.

User model creation User models, also called templates, are created from training feature sets to obtain a generic representation of a user that will be used for future comparisons. Many algorithms and procedures can be used depending on feature type and model family. For

speech or signatures this can involve training Gaussian mixture models (GMMs) using an iterative procedure.

Background model creation A background model, also called world model or anti-model, is needed by some biometric algorithms to provide normalisation for user presentation scores. They represent an "average" of the users from the population of the system. They are typically created by pooling together features of many different users.

Model storage Once their parameters are estimated, user models are stored in a secure location for use in later biometric operations.

Matching A biometric presentation is compared with a particular user's biometric model. This typically results in a presentation score which is somehow related to how likely it is that the particular user is the source of that presentation. This processing step varies depending on model and classifier types. For instance, GMM classifiers can use a likelihood-based score. For a given presentation, match scores are typically computed as the ratio of the score of the presentation with respect to a particular user's model to the score of the presentation with respect to the background model. Thus, this represents a kind of hypothesis testing, where the hypothesis can be phrased as "is it more likely that this presentation was produced by this particular user rather than anyone else in the background population?".

Threshold computation Several presentations belonging to a particular user and several presentations not belonging to that particular user (impostor presentations) are matched to that user's model to determine a hard limit (the threshold) below which a presentation will not be considered as belonging to the user. Thresholds can be user-independent (system-wide) or user-dependent, which is widely reported to give lower error rates. Again, many threshold computation procedures exist but most do work in the presentation score domain. Not all biometric modalities need a threshold.

1.2.2 Biometric operations using the processing steps

The processing steps described above are used in the following higher-level biometric *operations*.

Enrollment A user is added to the biometric system. A certain number of biometric presentation of a particular user are *acquired*, *preprocessed*, transformed into *features*, and *postprocessed*, then used to train a user *model* and adapt (retrain) the *world model* if necessary. The user model along with impostor presentations may be used to obtain a *threshold* for that user. The new user model is then *stored*, along with the threshold for that user if needed.

Verification The claim to a user's identity causes the presented biometric data to be compared against the claimed user's model. Thus, the biometric data is *acquired*, *preprocessed*, transformed into *features*, and *postprocessed*, before being *matched* with the claimed user's model and the resulting score being compared with the stored *threshold* computed for the claimed user or a generic *threshold* value.

Identification A database of user models is searched for the most likely source of the biometric presentation. Thus, the biometric data is *acquired*, *preprocessed*, transformed into *features*, and *postprocessed*, before being *matched* with all the user models of interest. The user model that obtains the highest score with respect to the presentation is suggested to be the source of the presentation.

In this thesis, we focus on biometric verification operations.

1.3 The problem of variability

In statistical pattern recognition, the optimal decision rule for a two-class problem such as biometric verification is given by the Bayes decision rule (also called Maximum A Posteriori (MAP) decision rule) in terms of posterior probabilities:

$$P(\omega_i|\mathbf{O}) > P(\omega_j|\mathbf{O}) \Rightarrow \mathbf{O} \in \omega_i. \tag{1.2}$$

Using Bayes rule, this can be reformulated in terms of priors and likelihoods:

$$\frac{P(\mathbf{O}|\omega_i)}{P(\mathbf{O}|\omega_j)} > \frac{P(\omega_i)}{P(\omega_j)} \Rightarrow \mathbf{O} \in \omega_i, \tag{1.3}$$

where the $P(\mathbf{O}|\omega.)$ terms are class-conditional likelihood functions, and the $P(\omega.)$ are the class priors. In this case, the score function referred to in Equation (1.1) is a likelihood ratio. If both classses are equiprobable ($P(\omega_i) = P(\omega_j)$), then the Bayes decision rule is known as the Maximum Likelihood decision rule and minimises the classification error.

In both cases, one of the assumptions typically made by the practioner* is that the training set used to learn the parameters of the class-conditional likelihood functions is a representative sample of the unseen test set on which the system will be used. If the test set data is distorted, the assumption no longer holds and the Bayes decision rule or Maximum Likelihood rules are no longer optimal.

In biometrics, the distortions of the data come from two main sources: intra-user variability, and changes in acquisition conditions.

1.3.1 Intra-user variability

Signature and speech are partly behavioural, partly physiological modalities. Emotional state and health factors come into play for both.

In speech processing, the emotional state of users is known to significantly alter the speech signal, including fundamental frequency and prosody [220], and in general to decrease speaker verification performance [156].

Many diseases, including the common cold, can change the voice of a person so much as to make it unrecognisable even by human listeners. Without going to this extreme, alcohol ingestion can also lead to significant changes in speech for some users, notably in fundamental frequency [127].

For signature, diseases such as Parkinson's can alter the writer's competence [42], and even coffeine absorption is known to affect psychomotor performance in writing tasks [306]. Furthermore, realisations of a signature by the same person, even in normal emotional and health conditions, always displays some alterations, especially the first time a specific writing instrument is used†.

Both speech and signature signals can be formalised as random variables resulting from discrete-time random processes. Since the signals are not deterministic, but stochastic, steady-state signal analysis techniques are not adequate for modelling. By using probabilistic methods instead, we can obtain models describing, amongst other, the amount of spread in the random variables, meaning that we can take into account the uncertainty on the realisation of the random variables (features) due to intra-user variability.

*others include independence of samples, knowledge of the parametric form of the underlying data-generating process, or sufficient training data

†Indeed, we have very frequently observed users commenting that "signature verification will never work for me, I never sign the same way twice"

1.3.2 Acquisition conditions

The conditions in which the signal is acquired is an important factor in its variability. While signature data acquired from a functioning sensor is virtually noise-free, speech data can suffer from several distortions:

Channel (convolutional) noise distorts speech signals as soon as they leave the speaker's mouth. All microphones have their specific transfer functions, most of the time non-linear, and reverberation in a room will also alter speech. It is known that speaker recognition performance degrades significantly when the enrollment and deployment channels are not matched. Noise robustness techniques used in speech processing for speech recognition (such as cepstral mean normalisation) can often be applied to speaker recognition. Compensation techniques derived from forensic speaker recognition [34] can also be applied to the biometric case.

Environmental (additive) noise is added to the speech signal by other audio sources surrounding the speaker, for example car noise, interfering speech, background music etc. In general, at low signal-to-noise ratios the error rates of speaker recognition systems drop significantly. Again, experience in other fields of speech processing can be drawn upon and applied to speaker recognition. It is generally observed that, except in extreme cases of channel degradations, convolutional noise has only a secondary effect on recognition performance compared to additive noise [267].

We posit that modelling information reflecting the acquisition conditions (signal quality measures) should be useful in improving the robustness of biometric verification systems to changes of data from the training conditions.

1.3.3 Motivations for the use of probabilistic models

In addition to offering a natural language for dealing with variability, using a probabilistic framework based on Bayesian networks for classification, classifier combination, and reliability estimation has several other advantages.

Firstly, the framework of Bayesian network has a very large expressive power. Both generative and discriminative methods of classification can be used, and efficient algorithms for both learning and inference are now widely available.

Secondly, at all levels, probabilistic outputs offer intuitive interpretability. This is not to be neglected in biometrics, where widespread development will see the technology end up in the hands of laypersons.

Thirdly, in the context of multiple classifier systems, having probabilistic output of base classifiers, coming from different models and modalities, simplifies the fusion task as no rescaling needs to take place.

Fourthly, the class imbalance problem, very common in biometrics where most attempts are from impostors, can be dealt with elegantly by learning distributions and changing the priors to reflect application-specific demands.

1.4 Objectives of the Thesis

We aim at developing models that can accomodate both the intra-user variability inherent in biometrics and the problem of changing acquisition conditions. This is achieved by resorting to probabilistic models throughout, from the base classifier level to the classifier fusion level.

Furthermore, the models developped should be genereal enough to apply to several biometric modalities, in the present case signature and speech.

At the level of base classifiers used in unimodal biometric verification, we develop probabilistic models to minimise the effects of intra-user variability.

We then research probabilistic models of classifier behaviour, taking into account factors that are detrimental to verification performance in single-classifier systems. If the quality measures reflect acquisition conditions, the aim is to obtain a reliability measure on the classifier's output which is to some degree robust with respect to changing environments. The system architecture is illustrated in Figure 1.2.

Research over the past 10-15 years has pointed undoubtedly to the fact that using multiple classifiers is one of the best ways to deal with failings of one of the classifiers. In multimodal biometrics, the implications are that if one modality is affected by either user variability or acquisition conditions, the other modalities should be able to compensate. Thus, we develop probabilistic models of classifier combination in the framework of Bayesian networks, and apply them to biometric verification tasks.

Finally, we aim at exploring the gains that can be obtained by incorporating quality measures in multiple classifier systems in a probabilistic manner, with the goal of outperforming state-of-the-art fusion algorithms operating on score data alone. The system architecture is illustrated in Figure 1.3.

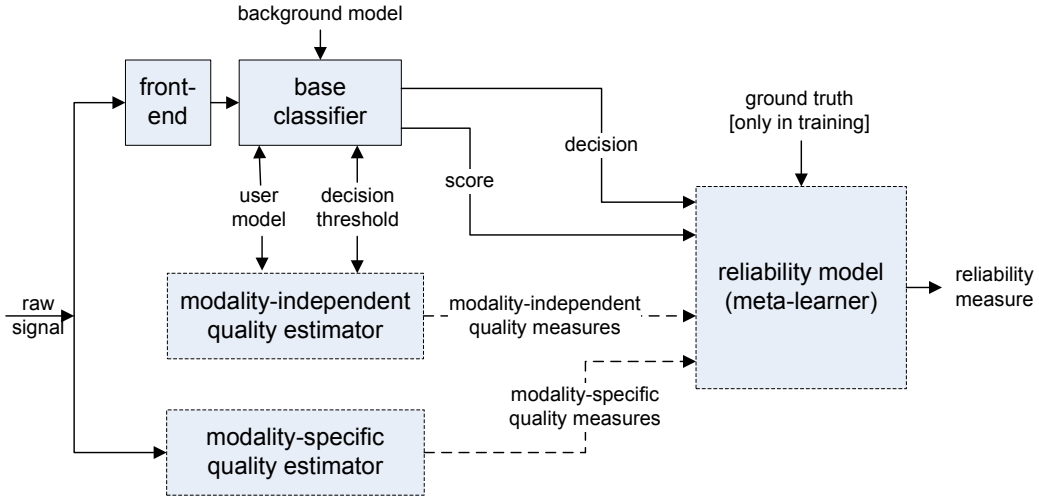


Figure 1.2 — Baseline verification system (plain lines) making use of quality measures through a reliability model (dashed lines)

1.5 Major Contributions

The major contributions of this thesis are:

In single-classifier systems:

1. The introduction of Gaussian mixture models for signature verification, in the framework of Bayesian networks, resulting in a state-of-the-art signature verification system.

2. The introduction of a Bayesian network-based reliability estimation model for single-classifier systems

In multiple-classifier systems:

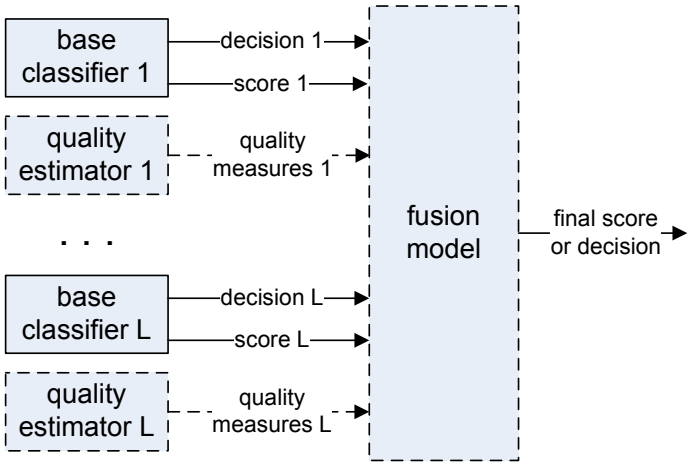


Figure 1.3 — multiple classifier verification system making use of quality measures through a fusion model (dashed lines)

1. The formalisation of different classifier combination algorithms as probabilistic models in the framework of Bayesian networks for both decision-level and score-level fusion
2. The introduction of new classifier combination models for biometric authentication based on Bayesian networks
3. The introduction of new probabilistic classifier combination models using quality measures for biometric authentication

In single- and multiple-classifier systems:

1. The development of new quality measures for speech and signature and the introduction of modality-independent quality measure
2. The development of a principled approach for the evaluation of quality measures
3. The proposal of a systematic view of quality measures in biometric authentication

1.6 Organisation of the thesis

The topics of the thesis chapters are represented graphically on the annotated multi-classifier biometric authentication system diagram of Figure 1.4.

Chapter 2 reviews the state of the art in the areas covered by this thesis, insisting on probabilistic models for unimodal speaker and signature verification, confidence estimation, and multi-classifier fusion methods based on confidence and quality measures.

Chapter 3 introduces the necessary theoretical background for the use of Bayesian network models in pattern recognition tasks.

Chapter 4 shows the theoretical basis for a Bayesian network approach to unimodal biometric classifiers, highlighting the differences and similarities between speaker verification and signature verification, and detailing the implementation of a signature verification classifier.

Chapter 5 proposes a taxonomy of quality measures and introduces new quality measures for use in speaker and signature verification.

Chapter 6 proposes probabilistic models of classifier behaviour that incorporate quality measures in order to perform reliability estimation in single-classifier systems.

Chapter 7 looks at classifier combination from a probabilistic perspective, showing how many common classifier combination schemes, both decision-level and score-level, can be interpreted and implemented as Bayesian networks, and introducing new fusion schemes based on Bayesian networks.

Chapter 8 proposes probabilistic models to combine quality measures in multiple classifier systems in order to improve verification performance over both the best baseline system and the best score-level fusion achievable without quality measures.

In closing, Chapter 9 offers some conclusions and points at some future work to be performed.

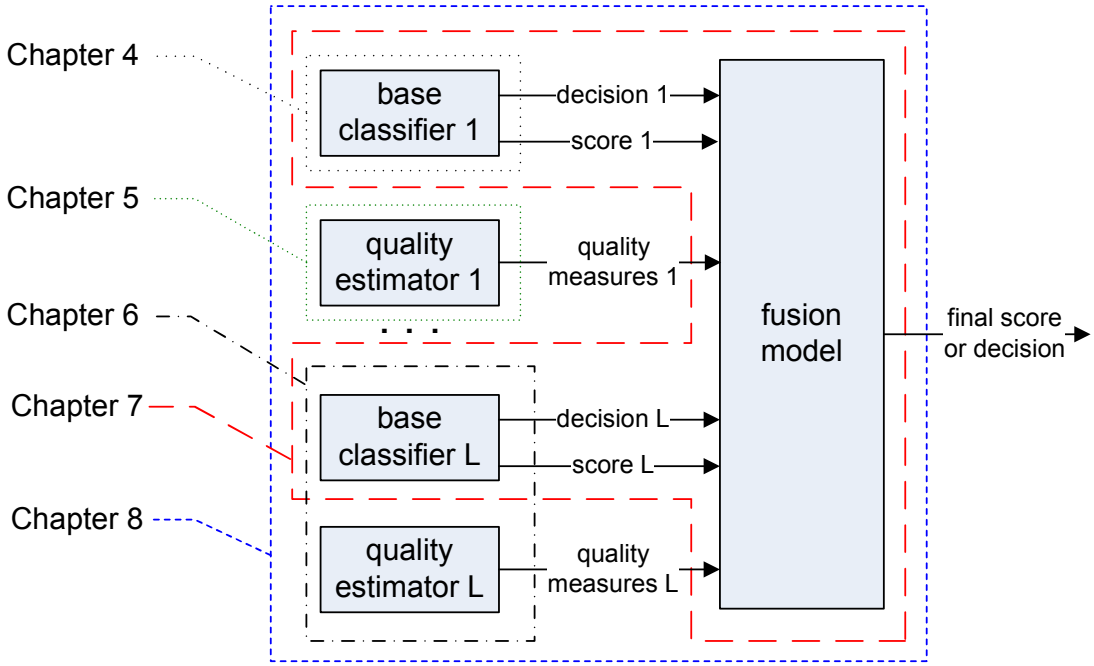


Figure 1.4 — Thesis plan as an annoted system diagram.

Part I

State of the art and background material

2 State of the art

2.1 Introduction

While biometric authentication has been the subject of ongoing research for at least 40 years, the relatively recent widespread availability of cheap computing power (meaning algorithms can be developed entirely in software), the emergence and theoretical maturing of the fields of machine learning and pattern recognition, combined with an increased interest in security have resulted in research on the topic becoming plethorical, with several dedicated conferences and most major signal processing or pattern recognition conferences now including sessions on biometric authentication. By necessity, in this chapter, we restrict ourselves to presenting classical methods and the most current approaches to specific issues in biometric authentication.

In Section 2.2, we present algorithmic approaches to speaker and signature verification, and expand on probabilistic methods in Section 2.3. Section 2.4 deals with the estimation of confidence in a verification result, reviewing approaches that have been used specifically in biometric authentication. Section 2.5 presents quality measures and their use in signature and speaker single-classifier biometric authnetication systems. In Section 2.6, we offer an overview of approaches to the combination of multiple classifiers, insisting on recent work dealing with confidence-dependent fusion and quality-dependent fusion. Finally, Section 2.7 discusses current tools used in the evaluation of biometric authentication systems, along with a presentation of commonly used datasets.

2.2 Single-classifier biometrics

2.2.1 Speaker verification

Automatic speaker recognition was pioneered in 1970 by Doddington [72], and subsequently became a very active research area. Today, speaker recognition systems and algorithms can be subdivided into two broad classes:

Text-dependent systems rely on the user pronouncing certain fixed utterances, which can be a

combination of digits, a password, or any other phrase. Thus, the user will prove her knowledge of the passphrase in addition to providing her biometrics. *text-prompted* systems are a special kind of text-dependent systems which ask the user to pronounce a certain utterance which may not be known in advance, to deter recording and replaying of the user's passphrase.

Text-independent systems allow the user to pronounce any utterance of their choosing.

Depending on the problem definition, several algorithms have been used to perform either text-dependent or text-independent speaker verification. We review four prominent approaches.

Dynamic time warping

Taking into account the dynamics of speech parameters has been proposed in the eighties and seen many subsequent refinments. A useful technique in this context is Dynamic Time Warping (DTW), which allows for compensation of the variability of speaking rate inherent to human speakers. Dynamic Time Warping has relatively low computational requirements, and is mostly used for text-dependent verification. Nowadays, DTW is less frequently used as a stand-alone speaker recognition algorithm [221], but rather as a way to supplement the decision process with auxiliary information. Recently, DTW has been used to model pitch contours as auxiliary information, providing improved recognition rates [2], and as part of a multi-model speaker recognition system [80].

Vector quantisation

Vector quantisation (VQ) for speaker recognition has been proposed and tested for a digit-based system over a 100-users database in 1985 [291], and has seen little use recently [193]. This approach is not commonly used anymore for speaker verification because it is consistently outperformed by statistical methods, which do take into account feature overlap and correlations by incorporating covariance information. However, VQ can outperform statistical methods when little data is available [189]. Yu et al. [328] have compared the hidden Markov model, dynamic time warping and vector quantisation approaches. Another comparison of vector quantisation and dynamic time warping is found in [11].

Neural networks

Neural networks have sometimes been used for text-independent speaker recognition, trained by providing both client and impostor data. Oglesby and Mason first proposed the a multi-layer perceptron (MLP) neural network with LPC-cepstral coefficients in 1988 and 1989 [210, 211], then expanded their work to a radial basis function network in 1991 [212] with better results than both VQ and MLP approaches. In [88], a radial basis function neural network is used for speaker identification on the TIMIT and NTIMIT databases. More recently, an auto-associative neural network has been tested on part of the NIST 2002 SRE database [113].

Support vector machines

Support vector machines, having been successfully been applied to many pattern recognition problems, have also been used in speaker recognition.

Schmidt and Gish proposed in 1996 to use support vector machines to perform speaker identification [282]. They tested their approach on a 26-user subset of the switchboard corpus and reported better results than with Gaussian Mixture models. In 2001, Gu and Thomas [112] reported improvements over GMMs by using SVMs for a 250-speakers phone-quality database. More recently,

Wan and Renals [309] have also reported better results for SVMs than for GMMs, and Louradour et al. [184] had similar results.

2.2.2 Signature verification

Over the past 30 years, numerous algorithms and models have been developed to verify on-line signatures*. While many algorithms rely on a temporal representation of the signature, some authors (notably Nalwa [202]) suggest that on-line signatures should be parameterized spatially. Currently, the lowest error rates are achieved by hidden Markov models using mixture of Gaussians output distribution, and Gaussian mixture models.

Dynamic Time Warping

The most widely studied on-line signature verification method is elastic matching (string matching) using dynamic time warping (DTW), also called dynamic programming (DP). Originally used in on-line signature verification by Sato and Kogure in 1982 [278], DTW has been gradually refined over the years. Two main approaches are seen in published literature: in the first the data points are used directly for matching after preprocessing (typically including subsampling), while in the second the signature is segmented according to perceived importance of boundary points.

Sakamoto et al. [273] have used position, pressure, and pen inclination to achieve 3% EER using three signature realisations templates per user with a 8-users corpus comprising a total of 1066 authentic signatures and 1921 forgeries. Jain et al. [135] have used a mixture of global features such as the number of strokes and local features, both spatial (e.g. x and y coordinate differences with respect to the previous point) and temporal (e.g. relative speed between points). They achieve about 2.2% EER using between three and five signature realisations templates per user with a 102-users corpus comprising a total of 1232 authentic signatures and 60 forgeries, thus probably underestimating the FAR.

Yanikoglu and Kholmatov [325], the winners of the signature verification competition 2004, have used a Dynamic Time Warping to align signatures based on two local features (Δx and Δy), after which they compute three distances with respect to that user's training set, perform PCA to decorrelate the three distances and classify on this last measure. They report 1.65% FRR and 1.28% FAR on a 94-users database, using 8 signatures per user for the user models and holding out 2 and 7 signatures for testing, thus testing with 182 authentic signatures. 313 skilled forgeries are used for testing.

Recently, DTW has been used as one of the classifier in a multi-classifier scheme [200]. It has also been used as the main classifier in a multi-stage verification system which was tested on 121 users with 726 authentic signatures and 89 forgeries, obtaining about 0.23% FAR at 3.63% FRR [239].

Faundez-Zanuy [81] has proposed intra-modal fusion using sum rule over the normalised outputs from a Vector Quantisation (VQ) classifier and a Dynamic Time Warping (DTW) classifier.

Neural networks

Neural networks have been explored for on-line signature verification but the performance reported in published literature is inferior to other methods such as DTW, HMMs or GMMs. Chang et al. [48] have used a Bayesian neural network trained with incremental learning vector quantisation. The EER achieved on chinese signatures is about 2.3%, using 4 signatures per user model . The 80-user corpus comprises a total of 800 authentic signatures and 200 skilled forgeries. Wu et al. [319]

* *Online* or *dynamic* signatures are digitised on-the-fly from an instrumented pen or writing surface, while *offline* signatures are written on ordinary paper and later digitised through digital imaging, typically via a scanner.

have used linear predictive cepstral coefficient derived from the x,y trajectory of the pen to train single-output multi-layer perceptrons (MLPs). Each "word" (chinese character) of a user's signature is modeled indpendently by an MLP. The EER achieved on chinese signatures is 4.3%, using an average of 12 authentic signature realisations and 12 forgeries to train each user's MLPs. The 27-users corpus comprises a total of 810 authentic signatures and 638 forgeries. It is not clear how this system would be applied to roman character-based signatures, where the relationship between the real letter and the signature-style letter is more ambiguous.

Euclidean distance

Euclidean distances or other distance measures have been used for on-line signature verification, generally achieving performance inferior to DTW, HMMs or GMMs. Rhee et al [249] use a model-based segmentation step prior to computing an Euclidean distance to a reference signature for each user. This results in an EER of 3.4%, using 10 signature realisations to build a reference signature with a 50-users corpus comprising a total of 1000 authentic signatures and 1000 very skilled forgeries.

Kashi et al. [144] have also used Euclidean distance with global and local features.

Regional correlation

The regional correlation approach has many proponents [222]. Nalwa [202] uses a function-based approach where the signature is parameterised using functions of arc-length, then cross-correlating these functions with each user's function prototype in her signature model. This achieves 3.6% EER on average over 3 different databases, amounting to a total of 204 users, 2676 authentic signatures and 1150 forgeries. Each user model was built using six signature realisations. While the corpus size is larger than what is used in most research papers, some pruning occured which caused some inconsistent genuine signatures to be rejected.

Lau, Yuen and Tang [177] have used a correlation-based approach and achieved about 1.7 % EER on a database of 100 persons, where each person contributes 5 authentic signatures and is forged 3 times. Each user is modelled using 2 signatures, thus resuling in testing with 300 authentic and 300 forged signatures.

2.3 Probabilistic models in single-classifier biometrics

Probabilistic methods rely on a parametric modelling of the signal. The modelling can be time-dependent (hidden Markov models, dynamic Bayesian networks) or not (Gaussian mixture model, Bayesian networks). The value of model parameters have to be learned from training data, which is a critical point in probabilistic models: sufficient training data has to be obtained. Probabilistic modelling has successfully been applied to both speaker and signature verification, in general outperforming other approaches.

2.3.1 Speaker verification

Hidden Markov Models

HMMs are very commonly used for text-dependent systems, where scores are typically obtained by finding the best path through the states. Ergodic (fully connected) HMMs have also been used for speaker recognition [189].

Poritz proposed using HMMs (5 states) to model speakers in 1982 [235], and performed identification on 10 speakers which resulted in no error. Rosenberg et al. [269] used speaker-dependent

whole word (digits) models, and tested on a 10-users population. Segmental HMMs (where states model events at a higher level than single vectors) have been used for both text-dependent and text-independent verification [182].

Gaussian Mixture Models

Schwartz, Roucos and Berouti first proposed probabilistic modelling for speaker recognition in 1982 [283]. In 1995, Reynolds [246] proposed using a mixture of Gaussian models (termed MoG or more commonly GMM) for modelling speech features belonging to a particular user. This approach has proved very successful and GMMs are now the dominant model for speaker recognition, often in combination with higher-level information provided for instance by DTW. A further refinement on the GMM method comes in the form of the universal background model (UBM) [247]: a large amount of data from many speakers is bundled together and a high-order GMM (typically 512 to 2048 mixture components [26]) is trained on that data. Then, a limited amount of speaker-specific data is used to adapt the UBM to each speaker. Essentially, the idea is to use a well-trained model (the UBM) as a good basis for initialisation of the user models. The vast majority of speaker recognition systems today are based on GMMs, and recent algorithms generally use a UBM-GMM system as a component. For example, a recent approach is to train a stacked SVM classifier on top of the vector of the means of the mixture components obtained by UBM-GMM training. This achieves substantial improvements in error rates [46, 65].

Bayesian networks

Sanchez-Soto et al. [274] have used Bayesian networks to model the interactions between pitch, energy, and two types of feature vectors in speaker verification. The network structure is learned by using K2 search with a BIC score (see Section 3.3.2).

Sanchez-Soto et al. [275] have further shown a model adaptation scheme based on adapting conditional probability tables.

2.3.2 Signature verification

Hidden Markov models

Inspired by the successful application of Hidden Markov models (HMMs) to on-line character recognition [172], HMMs have now become the best-performing models for on-line signature verification. The most commonly used similarity measure for HMMs is the log-likelihood ratio of the test signature given the user model to the test signature given the background model.

Yang et al. [324] have used quantised angle sequences as features, trying several HMM topologies and number of states. The best results, 3.8% EER, are obtained with a 6-states, left-to-right with skips topology, using 8 training signature realisations per model with a 31-users corpus comprising a total of 496 authentic signatures. The results are given for random forgeries.

Kashi et al. [143] have used a mixed-model approach, where global features such as average horizontal speed are combined with a variable duration discrete output HMM using inclination angles (with respect to the horizontal axis) and the difference between adjacent inclination angles as feature vectors. The reported error rate for a 20-states, left-to-right with no skips topology is 2.5% EER, using 6 training signature realisations per model with a 59-users corpus comprising a total of 542 authentic signatures and 325 forgeries.

Yoon et al. [327] transform the data into polar space with speed information, and further use vector quantisation to generate the feature vectors. A 5-states, left-to-right with skips HMM is used

for verification, resulting in 2.2% EER using 15 training signature realisations per model, with a 100-users corpus comprising 2000 signatures. The results are given for random forgeries.

Ortega-Garcia et al. [217] have used a 4-states, 8-components per state, left-to right HMM to model local features and apply score normalisation on the MCYT-50 subset, a model later varied [83] to a 2-state, 32-mixtures per state, left-to-right HMM, to obtain 5.79% EER on the SVC2004 development set (one of the best performance on this 40 users, 20 authentic and 20 forged signatures per user, training with 5 signatures).

These results show that signature verification algorithms based on HMMs have the potential to perform as well or better than those based on DTW or a variant thereof.

Bayesian networks

Xiao and Leedham [321] have used Bayesian networks to model relationships between off-line signature components, and outperformed a naïve Bayes model.

2.4 Confidence estimation

After a verification result has been obtained from a single classifier, it is often of practical importance to be able to know certain we can be that the algorithm did indeed provide the correct answer. To this end, a *confidence measure* on the classifier's output can be used. A *confidence measure* is a number quantifying the degree of trust that should be granted to a classifier's output (hard or soft). It is always based on a model (explicit or not) of normal operation for the classifier.

This problem of "knowing when the classifier is right", has seen numerous incarnations in very diverse areas and applications of pattern recognition, of which we will give but a few examples here. We focus on biometric applications in Section 2.4.2 and Section 2.4.3.

In handwriting recognition, Pitrelli and Perrone [229] have explored a number of confidence measures, the best of which encode a measure of the dispersion in scores of the top candidate words (in which case the confidence measures are for instance negative entropy, selectivity, or score ratios between candidates). Koerich [158] have studied rejection strategies based on confidence measures.

In medical decision support, Hamilton-Wright and Stashuk [115] use fuzzy inference and top candidate class score, as well as the difference between the top candidate score and the next candidate score (which can be interpreted as an assertion of support for the suggested class label). These features are modelled as an histogram and are used to compute confidence for human consumption. Also in the field of medical decisions, Neapolitan [206] has a Bayesian approach to estimate confidence in the output of an influence diagram, considering that the probabilities specified in the model are themselves uncertain .

Several other applications areas of confidence estimation exist, such as fault detection [181] or face recognition/segmentation [134].

2.4.1 Domain of evidence in confidence estimation

Confidence measures are typically based on data coming from a single domain; for instance, the output of the classifier itself is a primary source of information. The classifier output domain can be either continuous (generally called a soft output, or a score for probability density-based classifiers) or discrete (generally called hard output, label, or decision). In the case of biometric authentication, the classifier output domain is the most often used, as knowing the expected impostor and client score distributions provides important insights into the classifier's behaviour. Once the impostor and

client distributions are modelled, it will be possible to assign confidence values to different portions of the score range.

However, observing the classifier output alone is insufficient to understand the causes of errors, as a score deviating from the expected range might be caused by a large number of factors, including signal-domain noise, insufficient training data, or bad feature selection. The generic name we use for the data that is different from the classifier output is *quality measure*, a measurable indicator of a factor impacting the classifier behaviour (quality measures are the subject of Chapter 5). In contrast to *confidence* measures, we call *reliability* measures those that are inferred from a probabilistic model based not only on classifier outputs but also on quality measures. The term appears used in this exact sense in the works of Toyama and Horvitz [305], which devised a Bayesian network for estimating reliability of computer vision head pose estimation algorithms, using image features known to correlate with failures of each algorithm.

2.4.2 Confidence measures in speaker verification

Gaussian confidence measure

Bengio et al. [20] assume the client and impostor score distributions are Gaussians $\mathcal{N}(Sc; \mu., \sigma.)$. After estimating the parameters of the client score distributions (μ^c, σ^c) and those of the impostor distributions (μ^i, σ^i), the Gaussian confidence measures over a presentation score Sc is defined as:

$$CM_{Gauss}(Sc) = |\mathcal{N}(Sc; \mu^c, \sigma^c) - \mathcal{N}(Sc; \mu^i, \sigma^i)|. \tag{2.1}$$

They also propose a histogram-based density estimation method to alleviate the problem that the score densities may not be Gaussians.

Logistic confidence measure

The distribution of verification scores also serves as a basis for Nakasone and Beck [201]'s confidence measure. They propose a Bayesian confidence measure which can be expressed in speaker verification terms as the posterior probability that the utterance is from a client given the score:

$$P(\Omega = 1|Sc) = \frac{P(\Omega = 1)P(Sc|\Omega = 1)}{\sum_{\omega=0}^{\omega=1} P(\Omega = \omega)P(Sc|\Omega = \omega)}, \tag{2.2}$$

where ω represents either an impostor(ω=0) or a client (ω=1). By assuming that the client and impostor score distributions are Gaussian (which is often not true), they then define the confidence measure by fitting a logistic function to the posterior probability represented by Eq. (2.2):

$$CM_{Logistic}(Sc) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 Sc)}}, \tag{2.3}$$

where in our implementation the β exponential parameters are learned using a least-squares method. We adapt this measure from the forensic context by also computing $P(\Omega = 0|Sc)$ with a change of numerator in Eq. (2.2), then fitting a decreasing sigmoid $1 - CM_{logistic}$ to that posterior. This allows us to use this measure for the negative identification case also. One further change is needed since the ground truth is not available during testing. Thus, we replace Ω with the classifier's opinion CID, and use the appropriate measure at runtime depending on the classification result.

This measure presents two main drawbacks: it assumes Gaussian class-conditional distributions for scores, and does not take into account the actual error distributions of the classifier. These two drawbacks can also be seen as strong structural constraints that prevent overfitting and mean that this model may generalise better given a small amount of training data.

Nakasone and Beck's definition of confidence measure (Equation (2.2)) matches that of Fredouille et al. [87].

Bayesian confidence measure

In speaker identification, Gish and Schmidt [107] rely on the reasonable assumption that the scores of the top candidates in the case of correct classification is higher than those of incorrectly identified candidates. Their modelling is based on two distributions: The distribution of scores for incorrect classifications $P(Sc|DR = 0)$ (hereafter abbreviated $P_{wc}(Sc)$) and correct classifications $P(Sc|DR = 1)$* (hereafter abbreviated $P_{cc}(Sc)$). This is an interesting approach, since most confidence measures in speaker verification, and indeed in other fields of pattern recognition, are centred on the class-conditional distributions of scores, where the class of interest is the user identity Ω. They propose two methods to evaluate confidence in speaker identification applications, one based on significance testing, and the other on a Bayesian posterior probability $P(DR = 1|Sc)$.

The first, a significance-based confidence measure, can not be readily adapted to the verification case, because it essentially measures "how far on the tail of the distribution of incorrect classification scores the observed score occurs" , which is appropriate for identification, but not for verification. Indeed, while it can be expected and assumed that the mean of the $P_{wc}(Sc)$ distribution will be lower than the mean of the $P_{cc}(Sc)$ distribution, in verification the errors will be clustered around the threshold and correct decisions can be taken both below the threshold (correct reject) and above the threshold (correct accept).

Gish and Schmidt also propose a Bayesian confidence measure, which quantifies the posterior probability that the identification decision is correct given the score:

$$CM_{Bayes}(Sc) = P(DR = 1|Sc) = \frac{p_{cc}P_{cc}(Sc)}{p_{cc}P_{cc}(Sc) + p_{wc}P_{wc}(Sc)}, \tag{2.4}$$

where p_{cc} is the prior probability that the classification is correct, and p_{wc} is the prior probability that the classification is wrong. In identification, this can be estimated from results on an evaluation set. This measure can be applied in verification, but to set the priors an operating point must be chosen which corresponds to a particular threshold setting. An example for this is to choose the percentage of errors on an evaluation set; setting $p_{wc} = N(DR = 0)/N, p_{cc} = 1 - p_{wc}$ (where N is the total number of test cases in the evaluation set) ensures proper normalisation. For a well-performing speaker verification system, the ratio p_{cc}/p_{wc} is 10 or more. Thus, the confidence measure will be biased high and will most likely report high confidence. This can be compensated by using non-informative priors, meaning the confidence measure will be based only on the score distributions, without taking into account the priors. If the $P_{cc}(Sc)$ and $P_{wc}(Sc)$ score distributions were modelled as mixture distributions, this confidence measure should provide good accuracy when applied to verification tasks given that the score distributions are trained on an evaluation set which comes from an environment acoustically similar to that of the test set.

In general, a confidence measure based on the $P_{cc}(Sc)$ and $P_{wc}(Sc)$ distributions in verification needs to take into account the bimodal nature of the correct decision score distributions. This point is illustrated in Fig. 2.4.2.

Margin confidence measure

Poh and Bengio [232] use the false reject rate for a certain score (taken as threshold) subtracted from

*It should be noted that in the identification context Sc is an identification score, not a log-likelihood ratio as used in verification. Also, the semantics of DR change to $DR = 1$ if the candidate corresponding to the top identification score is indeed the target, and $DR = 0$ otherwise.

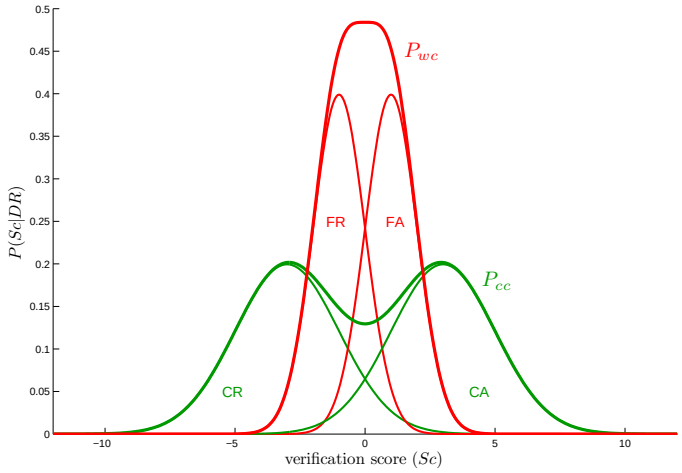


Figure 2.1 — idealised graph of correct verification ($P_{cc}(Sc)$) and verification error ($P_{wc}(Sc)$) score distributions showing the four sub-distributions: correct reject (CR), false reject (FR), false accept (FA), and correct accept (CA). Note that in reality the sub-distributions are likely to be non-Gaussian and overlap in a different way.

the false accept rate for the same threshold. Thus, the closer the score is to the decision threshold, the lower the confidence:

$$CM_{Margin}(Sc) = |FAR(Sc) - FRR(Sc)| \tag{2.5}$$

The client and impostor distributions are trained on an evaluation set. To avoid condition mismatch leading to erroneous test results, these functions can be trained on an evaluation set using conditions similar to those present in test conditions. This approach is interesting because it takes into account the distribution of errors with respect to a score, and it is quite generic: the sources of noise (both additive and convolutional) are subsumed and abstracted by their effects on the score distributions. This is far less complex than trying to model noise and distortions in the signal domain. The authors then show a theoretical framework for combining this confidence measure with a speech quality measure in order to enhance fusion in multimodal biometrics.

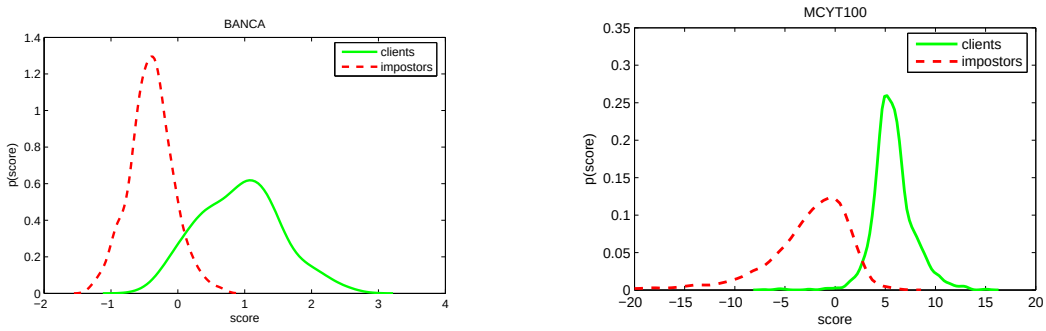
Issues in current approaches to confidence measures

These approaches to confidence estimation suffer from the same drawbacks, namely that the modelling assumptions are simplistic. The form of the probability densities are generally too simple (except for the CM_{Margin}, which is based on a smoothed kernel density estimator). The setting of prior probabilities is either not taken into account, or it can be done only for one of the two important prior probabilities: the prior probability of being an impostor or client, and the prior probability of the classifier having made an error. Lastly, the influence of signal quality is only taken into account through explicit modelling of noisy scores, and no attempt is made to try and refine the modelling into the causes of score variations. In Section 6 we propose the reliability model to correct for these deficiencies.

2.4.3 Confidence measures in signature verification

Confidence measures are commonly used in on-line and off line handwriting recognition. As an example, a class-dependent measure of the probability of one class being recognised correctly is described in [21]. However, to the best of our knowledge no confidence approach has been applied in the case of signature verification. We show the condifence measure methods used in speaker verification can readily be transposed to the case of signature verification.

If the classifier for signature verification is based on log-likelihood ratios, the semantics of the Sc term in the confidence measures for speaker verification need not change. Thus, all confidence measures based on the classifier output-domain in speaker verification are equally applicable to the case of signature verification. However, in doing so, and as shown on Fig. 2.2, care must be taken with assumptions about the relative variances of client and impostor score distributions: in signature verification, client distributions have typically smaller variances than in speaker verification. Furthermore, and as is the case in speaker verification, the Gaussianity assumption for impostor and client scores distributions does not seem to hold for signature verification either. Thus, all confidence measures using this assumption will suffer from the same problems.



(a) Speaker verification score distributions on BANCA database group 1

(b) Signature verification score distributions on MCYT100 database

Figure 2.2 — Speaker and signature verification system class-conditional classifier output distributions

2.5 Use of quality measures in single-classifier biometric authentication

Quality measures have received increased attention in biometric recognition in the past years. There is an increasing interest in generalising their use and pushing their incorporation in international standards*, as evidentiated by the NIST biometric quality workshop 2006.

Most research to date has concentrated on signal quality, specifically in the area of fingerprints and faces.

In fingerprint recognition, quality measures are quite mature, with several quality measures reaching near-standard status. The NIST finger image quality measure (NFIQ) [208] is one of the most widely used, and others such as FIQM (Finger Image Quality Measurement) or ENM (Equivalent Number of Minutia) are also prevalent in studies of fingerprint recognition using quality.

In face recognition, several quality measures have been proposed, such as brightness, exposure, focus, resolution, presence of glasses, "faceness", contrast [153], correlation with average face tem-

*e.g. ANSI/INCITS 379 and ISO/IEC 19794-6 for iris data, ISO 19794-5 for face

plates [165]. Some are under consideration for becoming international standards, such as colour balance, and lighting uniformity [59].

For the iris modality, commercial products routinely incorporate acquisition quality measurement, typically measuring the amount of information that can be extracted from the visible area of the iris. Academic research is also active, with new quality measures being proposed such as occlusion, motion blur, defocus blur, lighting, pixel counts, specular reflection and off-angle [281].

However, speech quality measures have not received as much attention in the context of biometric recognition. Likewise, very few signature quality measures have been developped. In the next sections we review existing approaches to estimating quality in speaker and signature verification.

2.5.1 Quality measures in speaker verification

In speaker verification, degraded acquisition conditions resulting in additive noise or channel noise cause utterance-dependent errors. Broadly, attempts to account for variability due to noise have used two approaches: either an explicit measure of signal quality is computed and modelled, or some other data transformation is applied without explicitely computing a quality measure.

Existing quality measures for speaker verification

The NIST Speech SNR Measurement [209] is based on sequentially fitting zero-mean Gaussian mixtures with different number of components to the signal. If a single Gaussian fits the signal well, it is deemed that no speech is present. noise estimate. If two Gaussian components fit the signal better, it is estimated that the signal comprises speech in high noise. Lastly, if a three-components model is the best fit, the signal quality is estimated good. The SNR is computed as a ratio of standard deviation of the fitted mixtures.

The U.S. Air Force research laboratory forensic automatic speech recognition SNR estimator [117] is based on histograms, and the SNR is estimated from noise/speech frames variance.

Huggins and Grieco [133] have also used an SNR measure as part of a more complex model of speaker identification. Additionally, they have proposed to use an estimate of the "amount of overlap" between the user models and the impostor models in feature space as a quality measure, with vector quantisation and Gaussian mixture models.

Many systems include a channel detector in order to switch or adapt user models to channel conditions in order to reduce mismatch between training and testing conditions [277, 301].

Other quantities not directly related to additive or convulutional noise can be used, for instance a quality measure based on deviations from a pitch model is proposed in [97].

Explicit use of quality measures in speaker verification

Score and quality modelling One approach to improving the robustness of the system under noisy conditions is to build explicit models of scores under certain degraded conditions by incorporating explicitely a quality measures in the model [45, 117]. Garcia-Romero et al. [97] have used a pitch-derived quality measure to scale likelihoods of a UBM-GMM classifier.

Missing feature theory In the missing feature approach, an explicit quality measure is computed on the elements of an acoustic feature vector, and elements that are below a certain quality criteria can be subjected to statistical treatement such as marginalisation or imputation[9, 78].

Other approaches to robustness in speaker verification

There are many approaches to handle mismatch and afford robustness to the classification, many inspired by similar work in speeech recognition*, at all level of the pattern recognition chain:

Speech enhancement Many classical tools in signal processing such as Wiener filtering can and have been used to try and enhance the speech signal, either in the time domain or in the spectral domain. It has been reported that spectral subtraction-based method yield good results in speaker identification [215].

Robust features Another possibility is to use a speech parameterisation that is to some extent immune from noise. Originally developed for speech recognition, RASTA-PLP [122] is one of the best-performing feature set, and is based on perceptual modelling of the human auditory system.

Feature transformation The features themselves can be modified to try and suppress some noise, or to match the testing conditions. One of the simplest, yet effective, techniques is cepstral mean subtraction [96], by which the stream of cepstral vectors is zero-meaned, with the effect of attenuating channel distrortions.

One approach proposed by Pelecanos and Sridharan [227] is feature compensation, whereby features are transformed using some warping function to conform to an expected distribution: for instance, the short-term distribution of feature vectors can be conformed to a Gaussian distribution. Pelecanos et al. [226] have more recently proposed to use a mixture of transformation matrices to match testing features back to conditions equivalent to the training environment.

Multi-condition training It is also possible to train the system directly in noisy conditions, as this is known to improve performance significantly [213]. However, this is not adapted for dynamically changing environments, where the noise level may not be the same from one utterance to the next.

A more sophisticated approach was proposed by Teunen et al. [301]: in speaker model synthesis, several speaker models are trained in different conditions, but new speaker models can be "synthesized" at test time by transforming existing models, thus reducing mimatch.

Score normalisation Score normalisation modifies the distribution of verification scores to improve the robustness of the system. It is one of the most widely used robustness techniques in speaker verification.

The most commonly used approach consists in normalising the score obtained on the user model by the score obtained on the background model; the idea being that the condition mismatch will affect both models and thus compensate for user model score drift [248, 268].

A refinement on this idea is to normalise the score by different background models for each condition[119].

With Auckenthaler et al. [12]'s Tnorm method, test presentations are scored against a set of impostor models, from which a variance and mean parameters can be trained. The presentation score given the claimed model is then normalised via the trained parameters. The same type of normalisation can be performed online and give at least equivalent results [130].

Other approaches are compared in [332].

*See for instance the review by Rose [267]

It has been reported that filtering or compensation approaches afford only limited robustness compared to score normalisation techniques [195].

2.5.2 Quality measures in signature verification

Existing quality measures in signature verification focus on the intra-user variability of the signature. Most of the results found in the literature are proposed for signature classifiers using non-probabilistic models such as Dynamic Time Warping (DTW), and are meant to address the deficiencies of the classifiers.

[36] have proposed the index of dissimilarity to measure the intra-variability of a signer. It consists in a normalised version of the DTW matching score between all enrollment signatures. They also propose a difficulty coefficient, which reflects how difficult a signature is to forge, based on the number of strokes, their duration, and the velocity of angle changes.

Dimauro et al. [70] have proposed a measure of local stability based on computing the number of points that correspond one-to-one in DTW: more 1:1 matches means higher stability. They then combine the stability information with the matching distance information via the product rule.

We find that is is more elegant and efficient to model intra-user variability directly, using probabilistic models, rather than attempt to correct for the deficiencies of matching algorithms at a later stage.

2.6 Multi-classifier and multimodal biometrics

2.6.1 Fusion levels

Multiple classifiers can be fused at several levels, corresponding to the 1.1:

Signal-level Signal-level combination can be used in multimodal systems or in multi-sensor systems. In this case, raw signals are combined and formed into feature vectors, which are then further processed. Many difficulties arise in multimodal signal-level fusion, including asynchronicity of feature streams (for example speech and face) and mismatched signal dimensions. Interesting recent trends in signal processing such as multimodal dictionaries [196] are starting to yield viable approaches for applications in biometrics, but so far signal-level fusion is not yet widespread in biometric authentication.

Feature-level Similar to signal-level combination, this consists in combining the features extracted from biometric traits into a unique features vector, thus yielding the same difficulties. The main differences are that in general dimensionality has been reduced*, and (in general) non-linear processing applied.

Score-level Score-level combination works on continuous random variables output from base classifiers. Depending on the fusion model, the scores may need to be normalised in order to belong to a common domain before the combination [270]. Score-level is very widely used in combination.

Decision-level This fusion mode consists in combining the decisions (hard labels) taken by each base classifier to obtain a final decision. This is often used in situations where no other data is available, or little training data is available, as it generally performs below score-level combination.

*This is not necessarily the case in signature verification (Section 4.5)

Performing fusion earlier in the processing chain is generally thought to provide more performance gains [271]. On the other hand, by moving higher up the processing chain, the random variables involved become more generic, and more general methods can be devised. Since one of the goals of this thesis is to develop methods that are reusable for different modalities, we focus on performing fusion at the score and decision level.

2.6.2 Fusion methods

Roughly speaking, fusion methods can be divided into fixed rules and trained rules [266], or equivalently into trainable and non-trainable ensembles [171].

Fixed rules

The main fixed rules for score-level fusion are the sum/mean rule, the product rule, and order statistics such as the maximum or minimum operators. For decision-level fusion, majority voting is one of the best performers, oftentimes outperforming more sophisticated, trained methods.

Fixed rule do not need to be trained. However, while fixed rules themselves are non-parametric, when used for score-level fusion, the data that is fed to them has to be carefully normalised. For exemple, the max rule will not work if one classifier in the ensemble has an output dynamic range an order of magnitude above the rest of the classifiers of the ensemble. Likewise, the mean operator (sum rule) is only meaningful if the classifier outputs are comparable in magnitude. Therefore, fixed rules are very dependent upon the normalisation of the different score streams. The normalisation parameters, generally a mean and a variance, or a minimum and a maximum value, have to be learned on a development set over the whole population (global normalisation), the impostor population (impostor-centric normalisation) or the client population (client-centric normalisation) (see Section 2.5.1). Calling fixed rules non-parametric, while *stricto sensu* correct, should not hide the fact that there is still a need for upstream parameter estimation.

Trained methods

Trained methods for classifier combination are countless, as nearly any existing classifier can be used by considering the output of base classifiers as features (stacking approach). Amongst other sources, Kuncheva [171] provides a good overview of many different models usable for combining base classifiers.

Trained versions of the main fixed rules exist, in the form of weighted majority voting and weighted sum.

2.6.3 Bayesian networks for combining multiple classifiers

The combination of Bayesian networks themselves, rather than the use of Bayesian networks to combine other classifiers, is a more common topic, of which we give but two examples. Using a method called Markov chain Monte Carlo model combination, Madigan et al. [186] generate ensembles of diverse Bayesian networks by removing various arcs. [148] have developed mixtures of simple Bayesian networks for classification of time series.

While the use of Bayesian networks for generic multiple classifier combination has not been studied extensively to date, some interesting theoretical results have been obtained by Bilmes and Kirchhoff [25], who expressed the product and mean rules as Bayesian networks, and experimented with other non-standard architectures for classifier combination at the feature level.

Garg et al. [101] use a Bayesian network (which is in fact equivalent to a multinomial combiner, as we show in Section 7.3.2) to fuse decisions of different classifiers.

Most recently, Chindaro et al. [53] have used structure learning algorithms to learn the Bayesian network functional equivalent of various classifier combination rules.

Given the scarcity of results on this topic, there is a need to try and build a better theoretical understanding of the uses and limitations of Bayesian networks as general-purpose combiners.

Structure learning for Bayesian networks

The application of Bayesian network structure learning algorithms to the problem of classifier combination is likely to bring improvements over models specified manually, especially when the number of base classifier becomes large. To the best of our knowledge, no research so far has been done on learning Bayesian network structure for combining biometic authentication classifiers. In Section 3.3.2, we review several existing structure learning algorithms that could be applied for combining classifiers.

2.6.4 Confidence-dependent classifier fusion and selection

After computing a confidence measure on a classifier output, for instance using one of the methods presented in Section 2.4, it is possible to use that information as an additional input for classifier fusion and classifier selection.

Confidence-dependent fusion methods can be divided into those that assign the same confidence to all outputs from a given classifier (fixed confidence), and those that compute confidence on a presentation-by-presentation basis (adaptive confidence).

Fixed confidence fusion

In that sense, weighted majority voting and weighted sum fusion, where the weights are proportional to the error rate of each classifier on a test set, can be considered as fixed confidence-based fusion methods, since less weight is given to classifiers for which the correctness of the output is most uncertain.

Another example of fixed confidence, used in classifier selection this time, is found in [279], where the best-accuracy classifier found by cross-validation is used for the whole dataset

Adaptive confidence fusion

The counterpart is dynamic classifier selection [316], which is based on the notion that different classifiers are competent for different partitions of the feature space. The classifier that gets to label the sample is the one that has shown the highest competence (lowest error) on a validation set in the partition where the sample to be classified lies. The same principle is used in [214].

In [6], confidence (defined as the posterior probability of the class given the observation) is used to select between two classifiers arranged in a serial architecture: if the first classifier is below a certain confidence threshold, the classification is deferred to the second classifier. This is referred to as a cascade architecure.

More recently, Dutra et al. [77] have used two confidence measures to weight the output of each classifier, before fusing the results via the maximum (resulting in classifier selection) or majority voting rule.

Foggia et al. [86] have proposed an interesting extension to the original work by Cordella et al. [57], which produced adaptive confidence values based on score output. A "reliability" measure is defined in [86], as a function of the input features and the classifier decision, with the aim of learning

mappings between errors and configuration in the input feature space. We note that this measure is not a probability, and operates on different features; thus, while close in spirit to our own approach, it is not equivalent to what we propose in Chapter 6.

Confidence-dependent fusion in biometric authentication

Several algorithms incorporating a notion of confidence have been developed specifically in biometric authentication.

Bengio et al. [20] have proposed an approach based on confidence measures described in Section 2.4.2, as well as an estimate of "model adequacy" computed for each presentation. These confidence measures on the base classifier outputs are incorporated as additional features along with the base classifier scores themselves, and three fusion models are compared: a GMM, an SVM, and an MLP. The reported results show that the only confidence measure bringing some (marginal) improvement is the "model adequacy" measure. The probable reason for this small improvement is that the confidence on the score brings no new information about the score itself, thus resulting in the modelling of redundant information. In contrast, modelling quality measures such as those presented in Chapter 5 provides auxiliary information that informative about, but not redundant with the scores.

Erzin et al. [79] use a combination of the score for the top candidate and the distance to the second-best candidate to combine speech, face, and lip modalities for speaker identification in a cascade configuration: the order in which classifiers are used is based on the respective confidence given to each classifier. Note that this is a classic approach to confidence estimation in speech recognition [139], and that it is not appropriate for two-class problems such as verification.

Poh and Bengio [232] propose an original approach for the use of their margin confidence measure (see Section 2.4.2). Linear weighted fusion is performed, but the fusion weights are composite: they are themselves made of a linear combination between a margin term and an another weigthing term (estimated on a validation set for instance). The results show some improvement on XM2VTS, but again suffer from modellling only score-domain data.

Confidence in a decision has been used to perform classifier selection in face verification [272], with an ensemble of experts using different similarity measures. In this case the confidence is defined as equal to the probability of error, and estimated from the class- (in our terminology, Ω) and error-dependent (in our terminology, DR) distributions of scores modelled as single Gaussians.

In the speaker verification application of [331], the score coming from a second classifier using other features is fused according to a confidence measure. The confidence measure itself is a sigmoid-style transform of the ratio of log-likelihood ratios for the two classifiers.

2.6.5 Quality-dependent classifier fusion and selection

The same criticism holds for confidence-based fusion as for confidence measures modelling only classifier output: no mechanism is provided for incorporating quality information in the fusion mechanism. A more recent trend in biometric authentication is to make quality (dominantly in the form of modality-specific quality measures) part of the fusion, in general in the form of a weighting factor.

The simplest approach is to train the fusion classifier on score computed from noisy data, without explicitely taking the quality of the signal into account. Similar to the multi-condition training approach used in speaker verification is the method used by Garcia-Salicetti et al. [100], where both SVMs and the mean rule are used to fuse noisy speech and clean signature data. The major issue with this approach is that the performance on clean data the system trained on noisy data

is likely to be inferior, since in this case the training/testing conditions mismatch would subside. Another, more theoretical issue is that this approach does not take into account the varying degrees of dependence between the two classifiers depending on the signal noise level.

A related approach has been proposed in multiple-classifier speaker verification by Solewicz and Koppel [290], where 9 ensembles of 4 classifiers are trained each on 9 different combinations of acoustic conditions (clean, low noise, high noise) for training and fusion development data sets. Signal quality is estimated via the means and standard deviations of filterbank outputs. Then, channel detection based on this estimate of quality is used to select one of the 9 ensembles. The obvious problem with this approach is its complexity and lack of flexibility (a problem shared by the approach in [133]), which requires no less than 36 different training sets.

Fusion of speech and fingerprint using (hand-labeled) signal quality measures is shown in [22], resulting in classification improvement if the fingerprint signal quality is taken into account. A speech quality measure based on an explicit noise model is used to weight the contribution of a speech expert to a speech and face multimodal system, achieving good results in degraded acoustic conditions [276]. Fusion of fingerprint and speech making use of fingerprint quality measures with polynomial regression models achieved about 2% reduction in error rates compared to the baseline fusion method without quality measure [304]. While results are good, the approach is not fully automated since quality is hand-labelled; However any of the recent automatic fingerprint quality measure extractor could be used.

[15] have proposed a Bayesian network for fusing several fingerprint matchers basd on discrete quality measures, and another topology for fusing two modalities. They do not provide an analysis of the independence assumptions made in the model, and no justification is given for the model topology.

Fierrez-Aguilar et al. [84] have developed an ensemble-based approach to quality-dependent fusion for signature and fingerprints. Several SVMs are trained with examples weighted by data quality, and during fusion the output of the ensemble members is combined with weights dependent on quality.

Using several quality measures, this approach was adapted by Garcia-Romero et al. [98] to fuse the contributions of a speaker verification system based on spectral features and a speaker verification system based on higher-level phonetic information.

Fierrez-Aguilar et al. [85] have applied quality-based fusion to intramodal fusion of two fingerprint matchers. The automatically extracted quality measure is used to linearly weight the output of each matcher, giving more weight to the one that is assumed *a priori* to be more robust to certain image degradations. No learning is attempted on the fusion function itself.

Nandakumar et al. [203] use the likelihood ratio of joint densities of quality measures and scores as a way to improve fusion in iris and fingerprint combination. However, the assumptions made that classifiers are independent is not strictly correct (as explained in Section 7.4.5, we should rather talk about conditional independence).

Kittler et al. [153] have proposed using either scores and quality measures or a non-linear combination thereof (tensor product between scores and quality measures) to improve fusion. SVMs and GMMs are used to model these features, either independently for each classifier, jointly for each modality, or jointly for all classifiers irrespective of the modality. While good results are reported for the use of the tensor product features, we suggest that another effective way of dealing with the non-linear interaction between quality measures and scores is to train the fusion model directly, rather than transforming the feature space

Poh et al. [234] have proposed a Bayesian network to perform intra-modal fusion in face verification, using automatically extracted quality measures.

With the exception of [15] (using a Bayesian network), [234] (using a Bayesian network) and Kittler et al. [153] (using a Gaussian mixture model), not much attention has been focused on probabilistic models in quality-based fusion, and most research has focused on support vector machines. No attempt has been made at combining signature data with quality information either.

2.7 Evaluation

This section defines the current methods and datasets used to evaluate biometric authentication systems.

Standard statistical pattern recognition tools are commonly used in the evaluation of performance of biometric verification systems, since biometric authentication is a two-class problem. However, the vocabulary used can be specific to the field.

2.7.1 Numerical performance measures

The *accuracy* of a classifier over a set of T samples is defined as

$$acc = \frac{1}{T}\sum^{T} I(CID(\mathbf{O}_t) = \Omega(\mathbf{O}_t)), \tag{2.6}$$

where I is an indicator function having value 1 if its argument is true, $CID(\mathbf{O_t}$ is the classifier decision regarding sample $\mathbf{O}_t$), and $\Omega(\mathbf{O}_t))$ is the true class of the sample. It is also convenient to define it in terms of a confusion matrix, shown in Table 2.1.

		CID	
		0	1
Ω	0	CR	FA
	1	FR	CA

Table 2.1 — Confusion matrix used in biometric authentication. Ω is the ground truth (0 for impostors, 1 for clients), CID is the classifier's decision. CR is the number of impostor attempts that are Correctly Rejected, FA is the number of impostor attempts that are Falsely Accepted, FR is the number of client attempts that are Falsely Rejected, and CA is the number of client attempts that are Correctly Accepted.

In this case we have an equivalent definition of accuracy:

$$acc = \frac{CA + CR}{CA + CR + FA + FR}, \tag{2.7}$$

where with respect to Equation (2.6) we have $T = CA + CR + FA + FR$.

Note that the (CA, CR, FA, FR) figures are computed on *decisions*, which are obtained by applying a decision threshold τ, and should therefore be denoted $CA(\tau)$ etc. Likewise, the accuracy function should be $acc(\tau)$. We avoid this notation for simplicity, but will make clear when necessary how the threshold is computed.

The use of the confusion matrix allows us to define other frequently used performance measures. The False Accept Rate (FAR) and False Reject Rate (FRR) at a given decision threshold τ are defined as:

$$FAR(\tau) = \frac{FA}{FA + CR}, \qquad FRR(\tau) = \frac{FR}{FR + CA}. \tag{2.8}$$

From $FAR(\tau)$ and $FRR(\tau)$, many different aggregate measures can be computed. The *Half-Total Error Rate* ($HTER$) at a given threshold is defined as

$$HTER(\tau) = \frac{FAR(\tau) + FRR(\tau)}{2}. \tag{2.9}$$

The *Equal Error Rate* (EER) is computed at the threshold τ' for which $FAR(\tau') = FRR(\tau')$:

$$EER(\tau') = FAR(\tau') = FRR(\tau'). \tag{2.10}$$

However, since all the performance measures presented here (acc, $FAR(\tau)$, $FRR(\tau)$, $HTER(\tau)$, $EER(\tau)$) depend explicitely on the chosen threshold τ, it is often useful to have a graphical representation of the value of the $FAR(\tau)$ and $FRR(\tau)$ for different values of the threshold, to have a more complete picture of the system behaviour. Indeed, setting a high threshold may decrease the FAR by rejecting more attempts, but will simulataneously increase the FRR for the same reason.

2.7.2 Graphical representations

The *Receiver Operating Characteristic* (ROC) curve has be used with various definition in the biometric literature since its inception [299], but is essentially a two-dimensional plot of one type of error against the other, for example $FAR(\tau)$ against $FRR(\tau)$, or more commonly $FAR(\tau)$ against $1 - FRR(\tau)$ for all possible threshold settings. The goal of a good system is to have the curve as close to one of the (appropriate) corners as possible.

Another way to represent graphically the FAR and the FRR is to use a *Detection Error Trade-Off* (DET) curve [187]. The error rates are plotted on both axes on a logarithmic scale, and the relative performances of well-performing authentication systems can be distinguished better than on ROC curves.

Both ROC and DET curves have attracted criticism, as they represent "ideal" performance attainable only with an *a posteriori* threshold, and they do not give information on the statistical significance of differences between systems. Two recent methods have been developed to cope with the issues, the expected performance curve, and the method of confidence bands.

These have not yet gained widespread usage, and we resort mostly to DET curves in this thesis.

The expected performance curve

Bengio et al. [18, 19] note that using the results of biometric evaluation in the form of ROC curves to compare different authentication systems is potentially misleading, as the varying thresholds in this case are computed directly on the test set. Therefore, there is no guarantee that a threshold (trained on a development set) which appears to show superior performance for a system will also show superior performance for this system on an unseen data set.

To remedy this issue, they propose the expected performance curve (EPC). To compute it, first define a cost function that is a linear combination of FAR and FRR on a given dataset at a given threshold. The combination weight is called α. Then, at each value of α, representing a different tradeoff, compute the threshold that minimises the cost function on a development set. Lastly, compute an aggregate measure of FAR and FRR (for instance, the HTER) and plot it. An example of an EPC for a signature verification system is shown in Figure 2.3.

Confidence intervals can be computed and plotted for the EPC, using a sampling technique similar to that presented in [28].

This method is directly applicable to multimodal evaluations, as it is based on scores only. It is sufficient to feed the fused multimodal scores to the algorithm.

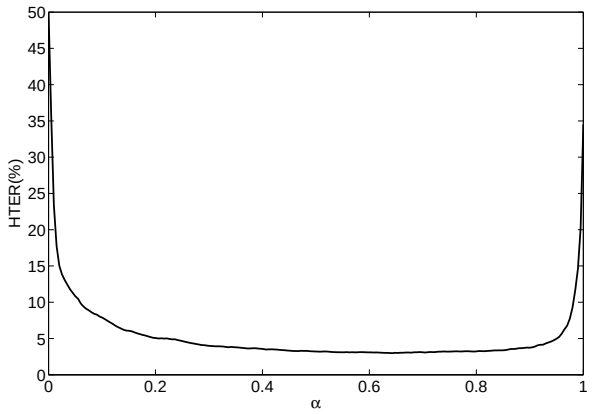


Figure 2.3 — Exemple of an expected performance curve for a signature verification system

Confidence bands

More recently, Dass et al. [60] developed an approach that gives confidence intervals over all possible threshold settings for a fingerprint verification system, as well as a specific computation to give the number of subject needed to achieve specified confidence intervals.

The approach involves computing matching scores within-user (both with same finger and a different finger from the original print), and between-user (impostor attempts). Then, multivariate Gaussian distributions are fitted and correlation matrices are computed. The fitted models are sampled a large number of times to generate artificial matching scores. Finally, assuming an asymptotically Gaussian model for score distributions, the confidence interval is computed at each FAR point in the range of interest, giving "confidence bands" on ROC curves that can be used to graphically assess the confidence interval over different operating points of the biometric system.

This approach has only been tested on fingerprints in the unimodal setting. While it seems it could be applicable to other modalities, applicability in the multimodal case is not established.

2.7.3 Application-oriented measures

Four other measures are mostly of interest in applied systems: the *failure to acquire rate*, the *failure to enroll rate*, the *time to enroll*, and the *time to match*.

The *failure to acquire rate* (FTA) is the percentage of users for which the system is unable to acquire a usable biometric sample during the enrollment and the transactions.

The *failure to enroll rate* (FTE) is the percentage of users for which the system is not able to generate a template to complete enrollment because of limitations of the technology or insufficient data quality.

The *time to enroll* (TTE) is the duration of the enrollment process from capture of biometric trait to the creation of the user template.

Lastly, the *time to match* (TTM) measures the duration of the matching process, from the end of the acquisition to the system's decision.

2.7.4 Databases

To enable comparison between biometric verification systems, it is necessary to test algorithms on well-known and widely available databases. While the situation in biometrics is not as advanced as

in other parts of machine learning, where (for instance) the UCI repository contains many databases in very different application domains, there are some frequently used benchmark databases.

Here, we concentrate on signature and speech databases.

Signature databases

Not many databases are publicly available to on-line signature verification researchers. Most research groups develop their own database. This is changing with international efforts such as the BioSecure Network of Excellence, which recently collected data for several hundreds of users. However, the final database is not available at the time of this writing.

BioMet [99] is a multimodal biometric database containing 5 modalities: speech, face, hand, fingerprint, and signature. The signature part consists of 3 sessions, the first using a standard non-writing pen on a pen tablet, and the other two using an inking per. The first session contains 5 genuine signatures and 5 forgeries executed by one forger. The second and third sessions contains the same material, with the addition of 1 forgeries executed by 1 different forger in each session. The total amount of data per subject is 15 genuine signatures and 17 forgeries.

MBioID [69] is a multimodal biometric database containing over 100 users collected in 2x2 sessions at least one month apart, where each user provides 10 signatures per session. Two scenarios are used to collect data on each acquisition day: one enrollment session where the user is sitting and can adjust the pen tablet orientation, an one transaction session where the user is standing up and cannot adjust the pen tablet orientation. Forgery data is currently random only, but skilled forgery acquisition will debut in the near future.

MyIdea [76] is a multimodal biometric database, whose signature part contains over 100 users recorded in three sessions. Each user provides 6 signatures per session. Each user is forgered 6 times by 3 different forgers, which are shown the offline version of the user. Additionally, an interesting aspect of this database is that 18 more forgeries are performed by forgers who are allowed to see a dynamic replay of the user's signature.

Philips Laboratories [73] contains 51 users, and each user provides 30 signatures. There are also 3000 amateur forgeries (practiced based on static image and over-the-shoulder), and 240 professional forgeries (contributed by forensic document examiners). This database is not generally available.

Detailed description of the MCYT-100, SVC2004, and BMEC2007 signature databases which we use for many experiments can be found in Section A.1.

Speech databases

Many databases are available freely or at low cost for speaker recognition tasks [44, 110, 191]. However, many are geared towards telephone-quality speech and thus not necessarily relevant to biometric vericiation over using a broadband microphone. Thus, we do not present PolyCost, SIVA, HTIMIT, LLHDB, Switchboard, OGI Speaker Recognition Corpus, YOHO, and others.

AHUMADA [219] contains speech for 104 male users and 80 females users, captured with 4 different microphones and more than 10 different telephone handsets, sampled on DAT tape. The data consists of isolated digits, strings of digits, phonetically balanced sentences, read text at various speaking rates, and spontaneous speech. The data was recorded in 6 different

sessions days or weeks apart. A subset of this data (electret microphone) can be used for initial testing of identity documents, but more users are needed.

BioMet 's speech part is acquired with a video camera and contains french material: digits, "yes", "no", and a phonetically balanced set of 12 sentences.

EUROM1 (the multilingual European speech database) [47] contains speech data in 7 languages (Danish, Dutch, English, French, German, Norwegian, and Swedish), with 60 users per language. Most users were only recorded once, but some were recorded on different days; this is not consistent from country to country The data contains numbers and read speech, with emphasis on phonetic balancing. The data is sampled at 20 kHz and quantised at 16 bits, and recordings take place in an anechoic room. Laryngograph data is also available for some subset of the data. While the total population is large, language effects may prevent this database from being used for speaker verification evaluation; furthermore, the inter-session time is not strictly controlled as this database was not origninally meant for speaker verification tasks.

King Speaker Verification (King-92) [124] contains speech data from 51 male users, recorded over 10 30-to-60 seconds sessions acquired weeks or months apart. The data is acquired with both telephone handsets (the recording quality varies depending on the location of the recording du to equipment differences) and a wideband microphone in a sound booth. The data is sampled at 8 kHz (originally 10 kHz in 1987 but resampled) and 16-bits quantised. This is not suitable for our identity documents purposes because the gender balance has to be representative of that found in the Swiss population, and the number of users is too limited.

MBioID 's speech part consists in 2x2 sessions of phonetically balanced sentences in french. The microphone used has a hypercardioid response pattern, low noise, and a very flat frequency response in the 20-20kHt range. It is connected via XLR cables to a high-quality pre-amplifier, the output of which is digitised by an external USB acquisition d. In the enrollment setting, the user's position with respect to the microphone is carefully controlled, the door and the window of the room are closed. In the transaction setting, the user is standing up and no instruction is given as to the distance or position from the microphone, and the window and door are opened.

STC Russian Speech Database [294] contains speech data from 89 users (54 males and 35 females), recorded over 15 or less 25-seconds sessions acquired within 1 to 3 months. The data is acquired using a high-quality, omnidirectional microphone in an office setting. The data contains 5 read sentences per session, is sampled at 11 kHz and quantised to 16 bits by a low-quality PC sound card. This can also be used as an initial development set for our application, though language effects may be a problem as the data is in Russian.

TIMIT Acoustic-Phonetic Continuous Speech Corpus [102] contains speech data for 630 users (438 males and 192 females), recordeded over a single 30-to-40 seconds session. The data is acquired in a sound booth, sampled at 16 kHz and quantised to 16 bits. The data contains phonetically-balanced read sentences in american English. The main problem with this data is that it has been recorded in one session and thus inter-session effects can not be evaluated. Furthermore, the amount of data per user is fairly limited. Campbell and Reynolds [44] advise against use of this database for speaker verification evaluation.

TSID Tactical Speaker Identification Speech Corpus contains 40 users (39 males, 1 female) recorded in a single session. The data is acquired in open air, using military radio handsets and an wideband electret microphone. The data contains phonetically-balanced read sentences,

digit strings, and spontaneous speech. This database is not suitable for our application because it is mono-session and gender-imbalanced. Furthermore, the open air environment does not correspond to the anticipated deployment environment.

Verivox [142] contains 50 male users recorded in a single 30-minutes session. The data is acquired in a sound booth, using a high-quality microphone. The data consists of digit sequences in Swedish. The data is sampled at 22 kHz, then downsampled to 8 kHz and quantised using 8 bits A-law companding. While a large amount of data is available per speaker, the gender-imbalance, Swedish language and narrowband sampling mean this database is not suitable for our purposes.

Detailed description of the BANCA and XM2VTS speech databases which we use for many experiments can be found in Section A.1.

2.8 Summary

In this Chapter we have reviewed unimodal classifiers applied to speaker and signature verification, insisting on probabilistic models that are commonly used in both modalities. This points out that, once features are extracted, the same tools can be used to model and verify signatures and speakers.

The output of unimodal classifiers can often be uncertain, a problem which is not specific to biometrics. Thus, research efforts have concentrated on developing confidence measures in order to gauge the level of trust that should be granted to a classifier's output. In general, confidence measures based on additional information, rather than only the classifier's output, should outperform those that do not. Several confidence measures have been developed for speaker recognition, and we reviewed their modelling assumptions. We also saw that very few confidence measures were developed for signature verification.

Confidence measures are generally thought to benefit from the inclusion of quality measures; however, many other approaches to handling environmental variability have been applied in speaker verification. The subject of quality measures in signature verification has generally not been researched extensively.

The combination of multple classifiers can be done according to a large variety of models, and including additional information on top of the classifier's output. We showed that Bayesian networks are one of the underused models for multiple classifier combination. Additional information brought in the fusion process is generally in the form of a confidence measure, or a quality measures. In the case of quality measures, significant performance improvements have been reported, while the use of confidence measures for fusion has generally not been so successful; we attribute this to the fact that a confidence measure is largely redundant with the score it models, whereas a quality measure brings additional information not generally available.

We finished the chapter by presenting evaluation methods and databases commonly used in biometrics research.

Bayesian networks: theoretical background

3

3.1 Introduction

Graphical models form a large family whose theoretical roots stem from a combination of graph theory with statistical theory. They were first applied to statistical physics and genetics [179], and are now seen as a unifying framework for a wide variety of probabilistic models. They provide a rich probabilistic language and come with a vast body of mature algorithms, which makes them particularly suitable and flexible for pattern recognition applications such as biometric verification.

Graphical models are subdivided into smaller model families depending on graph topology and semantics. Of special interest to us is the family of Bayesian networks, also called belief networks or influence diagrams. Bayesian networks represent probability distributions by using nodes to represent random variables, arcs between the nodes to represent dependence between random variables, and absence of arcs to indicate conditional independence between random variables.

In the next sections we will briefly mention the notions that are indispensible for the comprehension of the rest of the thesis. The interested reader is referred to two excellent books on Bayesian networks, one by Pearl [225], the other by Jensen [138]. Section 3.2 presents the basics of graph theory and how independence relationships are encoded in a graphical structure. Section 3.3 explains how parameter learning is achieved in graphical models, and Section 3.4 shows how inference can be performed. Section 3.5 reformulates the concepts presented in the more specific terms of an application to biometric authentication. Finally, Section 3.6 presents a summary of the chapter.

3.2 Graph theory and conditional independence

We start this section by precisely defining the concepts needed to describe Bayesian networks. The definitions are adapted from [17, 33, 138, 163, 179, 320].

3.2.1 Basic definitions

First we define the basic concept of graph:

Definition 1 (Graph) *A graph is a pair $\mathcal{G} = (V, E)$, where V is a finite set of vertices or nodes $V = \{V_1, V_2, \ldots, V_N\}$, and E is a finite set of edges or arcs (also called links or connections) between the nodes $E = \{E_{V_iV_j}, \ldots, E_{V_kV_l}\}$, with $E \subseteq V \times V$.*

The edges connecting the nodes between them can be of two different natures, which as will be seen later carry different semantics:

Definition 2 (Directed and undirected edges) *An edge $E_{V_iV_j}$ is undirected iff $E_{V_iV_j} \in E \Rightarrow E_{V_jV_i} \in E$, and directed iff $E_{V_iV_j} \in E \Rightarrow E_{V_jV_i} \notin E$. A directed edge from V_i to V_j can also be denoted $V_i \rightarrow V_j$.*

This distinction between directed and undirected edges allows us to broadly partition the family of graphical models into directed and undirected models:

Definition 3 (Directed and undirected graphs) *A graph is directed iff all edges in E are directed. It is undirected iff all edges in E are undirected. If the graph contains both directed and undirected edges, it is called a hybrid graph or chain graph.*

This distinction is one of the most fundamental in graphical models. Directed graphical models can be transformed to undirected graphical models and vice-versa, but with some restrictions as we will see in Section 3.2.5.

Definition 4 (Adjacency and neighbouring) *A node V_i is adjacent to (or equivalently, is a neighbour of) node V_j iff there is an edge between them. The adjacency of a node $Adj(V_i)$ is $\{V_j | E_{V_iV_j} \in E\}$.*

Based on this definition of adjacency, we can now define a special kind of edge, which will be useful to discuss inference algorithms in Section 3.4.

Definition 5 (Chord) *A chord is an edge between two non-adjacent nodes.*

3.2.2 Undirected graphs

The following definitions are fundamental in inference with Bayesian networks. As we will see in Section 3.4, these concepts are also used for inference in directed graphs.

Definition 6 (Complete graph) *An undirected graph or subgraph is called complete iff if there are edges between every possible pair of nodes in V (its set of edges is complete): $E = V \times V - \{E_{V_iV_i} | V_i \in V\}$*

Definition 7 (Clique) *A complete subgraph $\mathcal{G}_-$ is called a clique iff it is not a subgraph of another complete graph, that is if no superset of $\mathcal{G}_-$ exists that would be complete.*

This definition of clique is the one most often used in the literature on Bayesian networks and graphical models, and in some graph theory literature [116]. Some authors, however, refer to what we call a complete graph as a clique, while a clique is called a maximal clique [33, 289, 311].

Definition 8 (Separator set) *Let V_a denote the set of nodes comprised in clique $\mathcal{G}_a$, and V_b the set of nodes comprised in clique $\mathcal{G}_b$. The separator set S_{ab} is defined as $S_{ab} = V_a \cap V_b$.*

Definition 9 (Path) *A path is a series of nodes with intersecting adjacencies. More formally, a path of length k from V_1 to V_k is a sequence $P_{1k} = V_1, \ldots, V_k, k \geq 2$ of distinct nodes such that $E_{V_{i-1}V_i} \in E \forall i = 1, \ldots, k$.*

Definition 10 (Loop) *A loop is a path whose initial and final nodes coincide (a closed path).*

The presence of loops in the graphical model will condition the choice of inference algorithm.

Definition 11 (Connectedness) *A graph is connected iff for all nodes there exists at least one path reaching this node from any other node in the set of vertices, that is $\forall (V_i, V_j) \in V, \exists P_{ij}$. Otherwise, the graph is disconnected. A graph is singly-connected iff there exists only one path from any node to any other node, otherwise the graph is multiply-connected or loopy.*

Definition 12 (Tree) *A singly-connected graph is a tree.*

Definition 13 (Spanning tree) *A spanning tree for a graph has $|V| - 1$ edges forming a tree that are a subset of the fully connected graph.*

Fig. 3.1 shows an example of an undirected, multiply-connected graphical model that was used for estimating the pose (respective positions of limbs) of the human body [173]. Nodes are related to limbs.

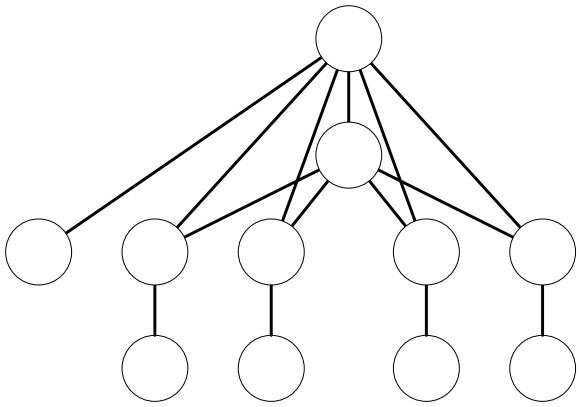


Figure 3.1 — Example of undirected graphical model for pose estimation [173]

3.2.3 Directed graphs

Definition 14 (Family: parents and children) *A node V_i is called the parent of node V_j iff $E_{V_iV_j} \in E$, where E is a set of directed edges, i.e. if there is a directed edge from node V_i to node V_j. Conversely, V_j is called a child of V_i. The set of nodes that are parents of V_j are denoted $pa(V_j)$. The family of a node is the nodeset consisting of the node itself and its parents.*

Definition 15 (Markov blanket) *The Markov blanket of a node V_i is the set comprising the parents, children, and the parents of the children of node V_i. In other words, it is composed by the union of the family of the node and the family of the children of the node, minus the node itself.*

Definition 16 (cycle) *A cycle is a directed path having the same start and end node. A graph that has no cycle is called acyclic.*

Definition 17 (directed simple trees and polytrees) *A directed tree is called simple if exactly one node (called the root node) has no parents, or equivalently if every node has at most one parent. Otherwise, it is called a polytree or forest.*

An example of a directed graph is given in the next section.

3.2.4 Bayesian networks

A Bayesian network is a directed acyclic graph where nodes have a 1:1 correspondence with the random variables in the domain $X_1, \ldots, X_N$, and each random variable has a conditional probability distribution given its parents $P(X_i|pa(X_i))$. In Bayesian networks, since the directed local Markov property holds (see Section 3.2.5), and assuming the nodes are numbered according to the topological order, the joint distribution over the variables of interest can be written as:

$$\begin{aligned} P(X_1, \ldots, X_N) &= P(X_1)P(X_2|X1)\cdots P(X_N|X_1, \ldots, X_{N-1}) \\ &= \prod_i^N P((X_i|pa(X_i)) \end{aligned} \tag{3.1}$$

Thus, they present a clear benefit in terms of interpretability and readability, as the factored form of the joint distribution can be read directly off the graph.

As an exemple of a directed acyclic graph, we will use a classic example of encoding of medical knowledge linking symptoms and possible causes, the so called "Asia" (or "chest clinic") Bayesian network [178]. The random variables corresponding to the nodes shown in Fig. 3.2 are all binary.

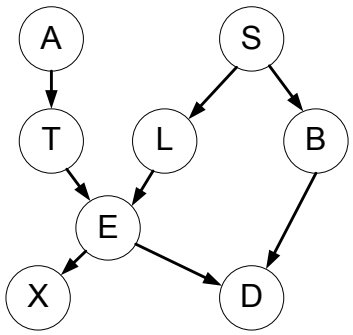


Figure 3.2 — The Asia Bayesian network $\mathcal{G}$

Some authors (e.g. [311]) contend that Bayesian networks should not bear this name, since learning and inference can be performed both by frequentist and Bayesian methods.

3.2.5 Independence and separation

Using Bayes' rule and the rules of probability arithmetic, joint probabilities of random variables can be decomposed into factors. For example, a simple joint probability of three random variables $P(A, B, C)$ can be written as $P(A|B, C)P(B|C)P(C)$. Thus, one approach to modelling a joint probability distribution requires specifying a number of probability distributions corresponding to the number of factors in the decomposition of the joint. The joint probability is then expressed as the product of the factors in the decomposition.

Each factor requires the specification of a number of values depending on the number of random variables it contains, and the cardinality of the domain of these random variables. For example, the factor $P(C)$ requires the specification of 1 value* if it is a binary variable, while the factor $P(A|B,C)$ will require us to set 4 values (again, the other 4 can be obtained by normalisation).

It appears that this approach to modelling is not tractable in the general case, where a large number of variables could be part of the joint probability distribution: for discrete models, the number of values that need to be specified grows exponentially in the number of variables.

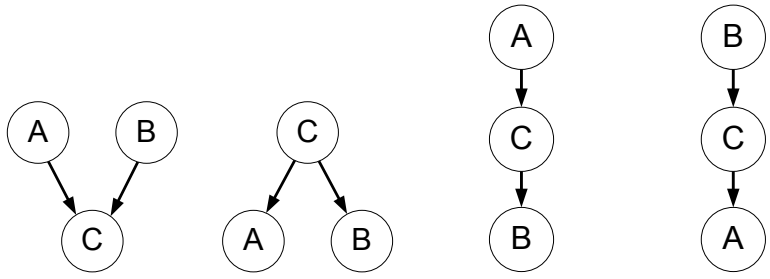
Therefore, in order to obtain an efficient representation, additional knowledge needs to be brought in to assume some independence between variables. This is a difficult task, one which can be achieved by applying human domain-specific expertise to the modelling, or algorithmically by data-based methods (see Section 3.3.1).

We first define independence between variables as [111]:

Definition 18 (Independence) *Two random variables A and B are independent iff $P(A,B) = P(A)P(B)$. Following Dawid's notation [62] this can be written $A \perp\!\!\!\perp B$.*

This is also called marginal independence, since the joint probability is a product of the two marginal distributions.

To illustrate, we will adapt an example from [17] with three random variables A, B, C, the joint distribution of which is modelled by a directed graph. Assume A represents the binary output of a speaker verification classifier, B represents the binary output of a signature classifier, and C represents a combined decision. If we make an independence assumption, we must remove a link between one pair of nodes. We remove the link between A and B, assuming the two biometric modalities are completely uncorrelated, giving the four possibilities depicted in Fig. 3.3.



(a) directed graph with convergent connection

(b) directed graph with divergent connection

(c) directed graph with serial connection

(d) directed graph with serial connection

Figure 3.3 — Possible directed graph topologies when assuming independence between A and B.

Working out the decomposition for the joint probability we obtain the following:

$$\begin{aligned}
\text{For graph 3.3(b):} \quad & P(C)P(A|C)P(B|C) = \tfrac{P(B,C)P(A,C)}{P(C)} = P(A|C)P(B,C) \\
\text{For graph 3.3(c):} \quad & P(A)P(C|A)P(B|C) = \tfrac{P(B,C)P(A,C)}{P(C)} = P(A|C)P(B,C) \\
\text{For graph 3.3(d):} \quad & P(B)P(C|B)P(A|C) = \tfrac{P(B,C)P(A,C)}{P(C)} = P(A|C)P(B,C)
\end{aligned} \tag{3.2}$$

*because the sum-to-one constraint on probability distributions (normalisation constraint) means that the other one can be obtained by simple subtraction. For example, $P(C = c_1) = 0.2$ implies $P(C = c_2) = 1 - P(C = c_1) = 0.8$. It also should be noted that for continuous random variables, the probability distribution can be represented by a classic parametric distribution such as a Beta or Gaussian density, in which case the number of needed parameters will decrease.

Therefore, these three topologies represent the same probability distribution. They encode the same conditional independence assumption: Given the value of C (called the conditioning variable), we have $P(A, B|C) = P(A|C)P(B|C)$. The definition of conditional independence is thus [62]:

Definition 19 (Conditional independence) *Two random variables A and B are conditionally independent given C iff $P(A, B|C) = P(A|C)P(B|C)$. This is written $A \perp\!\!\!\perp B \mid C$.*

So, in all three models 3.3(b)-3.3(d), we have $A \perp\!\!\!\perp B \mid C$, meaning if C is given (observed), A does not carry information about B and vice-versa.

As mentioned in Section 3.2.1, some directed acyclic graphs cannot be converted to undirected graphs with equivalent independence assumptions (indeed, the model shown in Fig. 3.3(a) is an example of this), and likewise some undirected graphs cannot be converted to an equivalent directed acyclic graph.

For simple graphs with a small number of edges, it is relatively easy to work out by hand from first principles which nodes are conditionally or marginally independent. For larger graphs, directional separation (see [224, Defs. 1.1 and 1.2]) gives precise rules to work out the conditional independence relations between random variables in any probability distributions represented by a directed graph.

D-separation (directional separation) can be defined in terms of blocked and active paths:

Definition 20 (Blocked and active paths) *A path is blocked by node V_b given a set of nodes B, if one of the following conditions apply:*

1. *$V_b \in B$ (it is observed) and V_b has one incoming and one outgoing arc (serial connection)*
2. *$V_b \in B$ and V_b has both arcs going out (diverging connection from V_b)*
3. *neither V_b nor any of its descendents is in B (they are not observed), and both arcs are incoming (converging connection on V_b)*

If none of these conditions apply, the path is said to be active.

Definition 21 (d-separation) *A set of node B d-separates to other sets of nodes X and Y if every path from a node in X to a node in Y is blocked given B. This can be written $\langle X|B|Y\rangle_{\mathcal{G}}$.*

D-separation can also be illustrated graphically, and efficient algorithms to test d-separation in graphs exist. For example, the Bayes ball algorithm [284] can compute a set of independent variables given a set of nodes and a conditioning set of nodes.

The rules of d-separation are consistent with a set of useful properties. We will only define one:

Definition 22 (directed local Markov property) *The directed local Markov property means that all variables are conditionally independent of their non-descendents given their parents. Another way of expressing this is to say that the joint probability distribution $P(V)$ is Markov to $\mathcal{G}$.*

The link between D-separation and independence proven by Pearl [225] is that if $\langle X|B|Y\rangle_{\mathcal{G}}$, then any distribution $P(V)$ that can be represented as a Bayesian network with DAG $\mathcal{G}$ has $X \perp\!\!\!\perp Y|B$, in which case we say $P(V)$ is Markov to $\mathcal{G}$. The converse ($X \perp\!\!\!\perp Y|B \Rightarrow \langle X|B|Y\rangle_{\mathcal{G}}$) means distribution $P(V)$ is *faithful* to $\mathcal{G}$ [52].

3.3 Learning algorithms for Bayesian networks

In this section, we will discuss algorithms that are used to train parameters of the conditional probability distributions associated with each node, as well as algorithms used to learn the topology of Bayesian networks. We will concentrate on frequentist approaches to learning, and only briefly touch on Bayesian methods.

3.3.1 Parameter learning

Assuming the topology of the Bayesian network is fixed, parameter learning can be divided into two cases: either training data is available for all random variables, in which case we say the data is fully observed, or it is not the case, and we say that the data is partially observed (or partially hidden). We will restrict the discussion to the case of multinomial distributions (for discrete data) and Gaussian distributions (for continuous data), as the combination of these two types of distributions according to the network topology is very expressive and largely sufficient for our needs.

Learning with no hidden variables

If all random variables in the training dataset are visible (including the class label), we can use a maximum likelihood (ML) approach, in which we view the parameters to be trained as quantities, the value of which we wish to learn. The best setting for the parameters is then taken to be that which maximises the probability of observing the training samples [74].

Assuming the training samples $\mathbf{o}_t$ drawn from the training set $\mathbf{O}$ are independent and identically distributed (vector-valued) random variables, we have the total likelihood of the training data given the model parameters $\mathbf{\Theta}$ on the Bayesian network defined by its graph $\mathcal{G}$ as:

$$P(\mathbf{O};\mathcal{G},\mathbf{\Theta}) = \prod_{t=1}^{T} P(\mathbf{o}_t;\mathcal{G},\mathbf{\Theta}), \tag{3.3}$$

where T is the number of training samples (vectors) in the training set. We usually take advantage of the fact that logarithm is a monotonically increasing function to transform the likelihood into a log-likelihood, an operation which makes subsequent steps easier since we only have to deal with sums:

$$LL = log\, P(\mathbf{O};\mathcal{G},\mathbf{\Theta}) = \sum_{t=1}^{T} log\, P(\mathbf{o}_t;\mathcal{G},\mathbf{\Theta}) \tag{3.4}$$

Using the directed local Markov property, we can now rewrite this as [198] :

$$LL = \sum_{n=1}^{N}\sum_{t=1}^{T} log\, P(V_i|pa(V_i),\mathbf{o}_t,\theta_i), \tag{3.5}$$

where N is the number of nodes in the network and θ_i is the parameter vector for node i. The nature of the parameters and the method for estimating them for each node will depend on the type of its distribution.

For multinomial distributions (discrete variables), the parameters to learn for each node θ_i directly correspond to the conditional probability distribution $P(V_i = k|pa(V_i) = j)$, that is the probability that this node V_i takes value k given that the parent has taken value j. This can be represented as a conditional probability table (CPT). The log-likelihood function in this case is

$$LL = \sum_{n=1}^{N}\sum_{t=1}^{T} log \prod_{j,k} P(V_i = k|pa(V_i) = j)^{I_{ijk}^t}, \tag{3.6}$$

where I_{ijk}^t is an indicator function (Kronecker delta) that is 1 iff the joint event $(V_i = k|pa(V_i) = j)$ happens in the training sample $\mathbf{o}_t$. Taking derivatives and using a Lagrange multiplier for normalisation, it can be shown [198] that the maximum likelihood estimate is then

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{\sum_{k'} N_{ijk'}}. \tag{3.7}$$

For Gaussian distributions (continuous variables), the log-likelihood for a continuous node V_i is given by

$$LL = log \prod_{t=1}^{T} \prod_{k=1}^{K} [\mathcal{N}(\mathbf{o}_{t_c}; \mu_{ik}, \mathbf{\Sigma}_{ik}, \mathbf{o}_t)]^{I_k^t}, \tag{3.8}$$

where K is the number of states the discrete parent can have, $\mathbf{o}_{t_c}$ is the subvector of $\mathbf{o}_t$ containing the continuous data for node V_i, μ_{ik} is the mean vector for node V_i if the discrete parent is in state k , $\mathbf{\Sigma}_{ik}$ is the covariance matrix for node V_i if the discrete parent is in state k, I_k^t is an indicator variable with value 1 if the discrete parent is in state i in training case $\mathbf{o}_t$.

The maximum likelihood estimates for the means and covariances, indexed by parent state k, are given by

$$\hat{\mu}_{ik} = \frac{\sum_T I_k^t \mathbf{o}_{t_c}}{\sum_T I_k^t} \tag{3.9}$$

$$\hat{\mathbf{\Sigma}}_{ik} = \frac{\sum_T I_k^t \mathbf{o}_{t_c} \mathbf{o}'_{t_c}}{\sum_T I_k^t} - \mu_{ik}\mu'_{ik} \tag{3.10}$$

Learning with hidden variables

If there are parameters to learn from hidden variables, the log-likelihood can be written as a combination of hidden variables and visible variables (those supplied by the training set):

$$LL = log\, P(\mathbf{O}, H; \mathcal{G}, \mathbf{\Theta}) = \sum_{t=1}^{T} log \sum_{h}^{H} P(h, \mathbf{o}_t; \mathcal{G}, \mathbf{\Theta}), \tag{3.11}$$

where H is the set of hidden variables.

In the fully observed case, the log-likelihood can be factored into a sum of local terms. However, if we have hidden variables we cannot decompose the complete likelihood into a sum of node-wise terms [198].

The expectation-maximisation (EM) algorithm [67] can be used to find a setting of model parameters corresponding to a local point of maximum likelihood. For brevity reasons we will omit the variational Bayesian treatment of EM, and restrict ourselves to the frequentist description. An overview of several variations of EM for graphical models is given in [91].

The overall idea of EM can be described thus: assuming we have some visible data $\mathbf{O}$ for which the log-likelihood of the probability density $P(\mathbf{O}; \mathcal{G}, \mathbf{\Theta})$ is difficult to maximise, if we can find another random variable H such that

$$P(\mathbf{O}; \mathcal{G}, \mathbf{\Theta}) = \sum_{H} P(\mathbf{O}, H; \mathcal{G}, \mathbf{\Theta}) \tag{3.12}$$

and that the log-likelihood of this new density is easy to maximise, then we will have an easier task. The end result we are interested in can be considered as the marginal of a model with a simpler likelihood function. What the EM algorithm does is provide meaningful estimates for these hidden data (E-step), maximise the log-likelihood (M-step), and repeat.

In the initialisation step the parameter priors are initialised to some value, for instance to a random value. A better prior can be found by using fast clustering algorithms such as k-means.

The first step (expectation step or E-step) is to compute the expectation over the missing data, treating the model parameters at the current iteration $\mathbf{\Theta}^i$ and the observed data $\mathbf{O}$ as fixed. The objective criterion is

$$J(\mathbf{\Theta}|\mathbf{\Theta}^i) = E_H[log\, P(\mathbf{O}, H; \mathbf{\Theta})|\mathbf{\Theta}^i, \mathbf{O}], \tag{3.13}$$

where E_H denotes the expectation taken over the hidden variables, and $\mathbf{\Theta}$ is a candidate set of model parameters.

The second step (maximisation step or M-step) is to find a setting of model parameters to maximise the objective criterion:

$$\mathbf{\Theta}^{i+1} = \underset{\mathbf{\Theta}}{\operatorname{argmax}}\, J(\mathbf{\Theta}|\mathbf{\Theta}^i) \tag{3.14}$$

The iterations can then be repeated until convergence, which can be established for instance by setting a threshold on relative log-likelihood increase between two iterations.

3.3.2 Structure learning

Since learning an unrestricted Bayesian networks topology from data is an NP-hard problem [121], practical structure learning algorithms impose certain restrictions on the problem, typically in the connectedness of the graph.

Structure learning for Bayesian networks can be seen as a typical optimisation problem, where a search algorithm is guided by a cost function towards better-scoring models.

Another view of structure learning is as an application of a set of rules specific to directed graphical models (such as d-separation), and the explicit computation of independence relationships between variables, for instance based on mutual information. An example of this approach is shown in [50].

Lastly, model topology can be chosen using human expert knowledge of the domain and good understanding of directional separation (good principles are given in [137, Chapter 2]).

Since a vast amount of literature has been published on stucture learning (among others [1, 51, 121]), in this section we will only briefly review cost functions and search methods that can be applied to Bayesian network structure learning.

Cost functions can take as input training data, a Bayesian network graph, and domain-specific information [51], and return a score indicating how well the topology indicated by the graph corresponds to the training data. Many of these cost functions (such as MDL [263]/BIC and AIC [5]) are not restricted to Bayesian networks models [265], while others such as BGe, BDe and BDEu [121] are specific to graphical models.

Search algorithms move in the space of model parameters by trying to maximise the cost function. Again, some algorithms that can be used to learn Bayesian networks such as Tabu search [109], hillclimbing, simulated annealing [149], and genetic search [176] are not restricted to graphical models, while other search algorithms such as K2 [56] (which keeps adding arcs while the cost function improves) are specific to Bayesian network models.

3.4 Inference in Bayesian networks

A trained Bayesian network, that is one for which the conditional probability functions have been learned for each node, can be used to perform inference. Inference procedures allow for computing the effects on other nodes of observing certain variables. Formally, probabilistic inference consists in computing a posterior distribution for a set of query variables, given observed variables.

A useful notion to introduce at this point is that of probability potential.

Definition 23 (Probability potential) *A probability potential (or potential for short) is a non-negative function defined over the domains of a set of random variables.*

Potentials are used in undirected graphical models to represent the "strength" of the link (or the degree of association) between neighbouring nodes. If the domains of the potential are discrete, it can be represented as a multinomial distribution. If the domains are continuous, a parametric distribution (typically a normal distribution) can be used. If the domains are both continuous and discrete, a linear conditional Gaussian distribution (mixture of Gaussians) can be used. A potential $\phi(X)$ can be converted into a probability distribution $P(X)$ by normalisation:

$$P(X) = \frac{\phi(X)}{\sum_X \phi(X)} \tag{3.15}$$

3.4.1 Variable and bucket elimination

The variable elimination algorithm (VE) proposed in [329] and the bucket elimination algorithm proposed in [64] operate on the factorised representation of joint probability density. The query of interest divides the set of variables V into three subsets of variables: query variables V_q, observed variables $\mathbf{O} \triangleq V_o$, and unobserved variables H. The VE algorithm uses the distributive law to break up the marginalisation on unobserved variables into small independent terms, thereby making the computation more efficient. This principle has a very large number of applications in many areas of engineering, and is known under the name of generalised distributive law [4].

For instance, we could compute $P(S|D, A)$ on the Asia Bayesian network of Fig. 3.2. In this case, the query variables set is $V_q = \{S\}$, observed variables are $V_o = \{D, A\}$, while the unobserved variables are $H = \{T, L, B, E, X\}$*. The trivial factorisation for this graph is

$$P(V) = P(A)P(S)P(T|A)P(L|S)P(B|S)P(E|T, L)P(X|E)P(D|E, B) \tag{3.16}$$

In order to obtain the posterior of interest $P(S|D, A)$, we need to marginalise over all the unobserved variables (except, as mentioned, the query variables):

$$\begin{aligned} P(S|D, A) &= \alpha \cdot \sum_H P(V, D = d, A = a) && (3.17)\\ &= \alpha \cdot \sum_{X,T,L,B,E} P(a)P(S)P(T|a)P(L|S)P(B|S) && (3.18)\\ &\quad \cdot P(E|T, L)P(X|E)P(d|E, B) && (3.19) \end{aligned}$$

where α is the normalisation constant stemming from the use of Bayes' rule to transform the joint marginal $P(S, D, A)$ into the conditional $P(S|D, A)$. Using the distributive law, the sums can be "pushed to the right" in order to be reused after their computation. We start by choosing an elimination ordering corresponding to the sequence of variables (X, T, L, B, E), which leads to distributing the sums as follows:

*strictly speaking, the query variables set is also unobserved, but for notational convenience we define $H \cap V_q = \emptyset$

$$P(S|D,A) = \alpha P(A=a)P(S)\sum_E\sum_B P(B|S)P(D=d|E,B)\sum_L P(L|S) \cdot \underbrace{\sum_T P(T|a)P(E|T,L)}_{\phi_T(E,L)}\underbrace{\sum_X P(X|E)}_{1} \tag{3.20}$$

The potential $\phi_T(E,L)$ has a domain corresponding to the domains of E and L, since T has been marginalised over. We can now carry over this potential to the next summation and continue in the same fashion until we are left with the desired answer:

$$P(S|D,A) = \alpha P(A=a)P(S)\sum_E\sum_B P(B|S)P(D=d|E,B)\underbrace{\sum_L P(L|S)\phi_T(E,L)}_{\phi_L(E)} \tag{3.21}$$

$$= \alpha P(A=a)P(S)\sum_E \phi_L(E)\underbrace{\sum_B P(B|S)P(D=d|E,B)}_{\phi_B(D=d,E)} \tag{3.22}$$

$$= \alpha P(A=a)P(S)\underbrace{\sum_E \phi_L(E)\phi_B(D=d,E)}_{\phi_E(D=d)} \tag{3.23}$$

$$= \alpha P(A=a)P(S)\phi_E(D=d) \tag{3.24}$$

The efficiency of bucket or variable elimination is extremely dependent on the order of elimination. Some heuristics have been proposed that give efficient orderings [155, 329].

3.4.2 The junction tree algorithm

The junction tree algorithm transforms directed graphical models into undirected equivalents, called join trees or junction trees, based on which inference can be performed. Before explaining the algorithm's operation, it is necessary to define a few notions.

Definition 24 (Moral graph) *The clique graph $\mathcal{G}_c$ of a graph $\mathcal{G}$ has nodes consisting of the cliques of $\mathcal{G}$ and edges joining any two cliques having a non-empty separator set. This is also called a cluster graph or junction graph.*

A moralised graph can be triangulated, an essential step in the construction of the junction tree.

Definition 25 (Triangulated graph) *An undirected graph is triangulated iff any cycle of length more than 4 (4-cycle) has a chord. This is equivalent to saying that its clique graph has a junction tree.*

Definition 26 (Clique graph) *The clique graph $\mathcal{G}_c$ of a graph $\mathcal{G}$ has nodes consisting of the cliques of $\mathcal{G}$ and edges joining any two cliques having a non-empty separator set. This is also called a cluster graph or junction graph*

For representation convenience and as is commonly done in Bayesian networks literature, we will introduce separator sets as additional annotation nodes between the corresponding cliques in clique

graphs. This representation allows us to make use of the fact that the factorisation of the joint distribution encoded by the Bayesian network is given by

$$P(V) = \frac{\prod_i \phi_{c_i}}{\prod_j \phi_{s_j}}, \tag{3.25}$$

where ϕ_{c_i} are the clique potentials, and ϕ_{s_j} the separator set potentials.

The use of nodes for separator sets also makes it easy to verify that the running intersection property is respected:

Definition 27 (Running intersection property) *A graph that satisfies the running intersection property is one in which, for each pair of clique nodes (V_i, V_j) all nodes on the path between V_i and V_j contain $V_i \cap V_j$.*

To derive a junction tree from any directed acyclic graph, the algorithmic procedure described in Algorithm 3.1 applies. A good description of practical implementations of the junction tree algorithm can be found in [129].

Algorithm 3.1 Junction tree algorithm (JTA)

1: Moralisation: from the original Bayesian network $\mathcal{G}$, obtain the moral graph $\mathcal{G}_m$
2: Triangulation: triangulate $\mathcal{G}_m$ to obtain $\mathcal{G}_t$.
3: Clique identification: identify the set of cliques $\mathcal{C}$ in $\mathcal{G}_t$.
4: Clique graph construction: build the clique graph $\mathcal{G}_c$ by adding separators between every cluster.
5: Junction tree construction: remove the unnecessary separators so that the resulting graph $\mathcal{G}_j$ (the junction tree) satisfies the running intersection property.

The moralisation step is achieved by dropping directionality on the arcs of the original Bayesian network $\mathcal{G}$, then for each node V_i add an edge (called moral arc) between each pair of nodes in $pa(V_i)$ if there is none. An example is given in Fig. 3.4(a) for the moral graph of the Asia Bayesian network: we married parents T and L as well as E and B, then removed the directionality of the edges of the orginal network in Fig. 3.2.

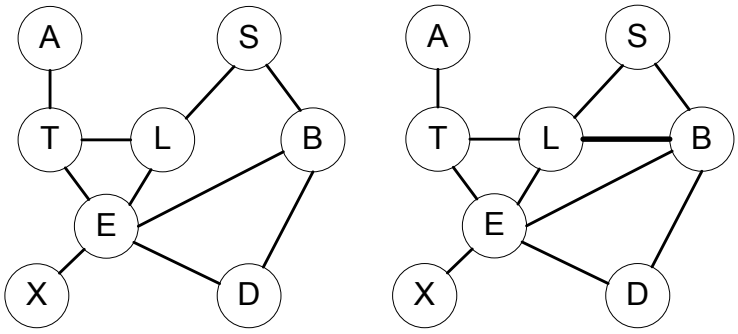
While finding an optimal triangulation is NP-complete, several algorithms exist to perform efficient triangulation giving good results in practice [155]. Still making use of the Asia network example, looking at Fig. 3.4(a) it appears that the moral graph is not triangulated because of the chordless cycle (S, L, E, B). Thus, we can introduce an additional edge between L and B to obtain a triangulated graph, shown on Fig. 3.4(b). This step guarantees that we will be able to find a clique graph with the running intersection property.

In the Asia network example, the small size of the graph means the cliques can be identified manually. They are represented graphically on Fig. 3.5(a), and the set of cliques in this case is

$$\mathcal{C} = \{\{E, L, T\}, \{B, L, S\}, \{D, E, B\}, \{B, L, E\}, \{A, T\}, \{X, E\}\} \tag{3.26}$$

The clique graph is constituted by merging clique member variables into larger clique nodes, and adding nodes containing separator sets between clique nodes. It can contain additional edges that are not essential to satistisfying the running intersection property. These are shown as dashed lines in Fig. 3.5(b). By removing all redundant edges and separators, we obtain the final junction tree shown in Fig. 3.6.

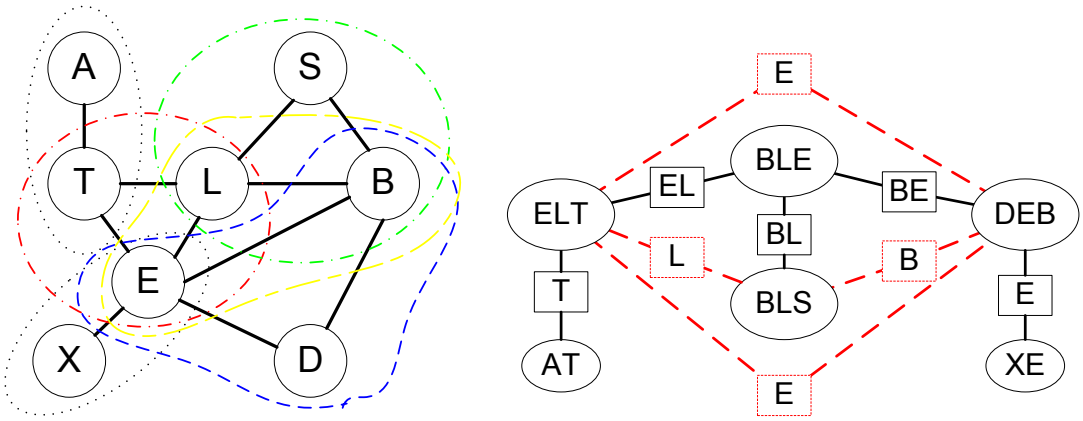
The junction tree can then be used together with a message passing algorithm to perform inference in an efficient manner.



(a) The moral graph $\mathcal{G}_m$ of the Asia Bayesian network $\mathcal{G}$

(b) The triangulation $\mathcal{G}_t$ of the moral graph $\mathcal{G}_m$

Figure 3.4 — Exemple of moralisation and triangulation step in the junction tree algorithm



(a) The cliques for the triangulation $\mathcal{G}_t$ of the Asia network

(b) The clique graph $\mathcal{G}_c$ for the triangulated graph $\mathcal{G}_t$. Square nodes indicate separator sets

Figure 3.5 — Exemple of cliques and clique graph step in the junction tree algorithm

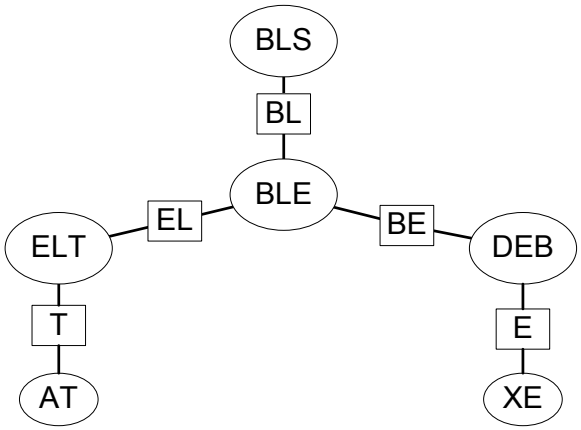


Figure 3.6 — Junction tree $\mathcal{G}_j$ for the Asia network

3.4.3 Message passing and belief propagation

We will restrict the exposition in this section to the undirected belief propagation scheme, since it can be used to perform inference on Bayesian networks once they have been transformed into a junction tree (see Section 3.4.2).

Undirected belief propagation works on non-loopy graphs, and is alternatively called message passing, the Hugin algorithm or clique tree propagation. In some cases message passing is taken to be part of the junction tree algorithm, whereas we consider the latter to be the algorithm used to transform a DAG into a junction tree. While several versions of the message passing protocol exist, we will focus on the two-phase "serial" version, whereby all marginal distributions can be computed in two phases on a tree.

The idea of message passing is to perform a series of computations such that at the end of the computation, the potentials associated with the nodes in the junction tree contain the marginal distributions over the domains of the nodes. Furthermore, a global consistency will be ensured.

Definition 28 (Global consistency) *A clique tree is globally consistent if, for any pair of clique nodes (C_i, C_j) with separator set S, we have*

$$\sum_{C_1 \setminus S} \phi(C_i) = \sum_{C_2 \setminus S} \phi(C_2) \tag{3.27}$$

Algorithm 3.2 Message passing on a junction tree

1: Initialise all clique potentials $\phi(C)$ to the corresponding term in the factored representation of the DAG
2: Initialise all separator potentials $\phi(S)$ to 1
3: Choose an arbitrary node as the root node
4: Perform the evidence collection message pass
5: Perform the evidence distribution message pass

To initialise the algorithm, each clique is associated with a potential having the same domain. If the underlying model is a DAG, as is the case if we apply message passing to a junction tree, we assign one or more terms in the factorisation of the Bayesian network to each potential with corresponding domain. The potentials corresponding to the separator sets can be initialised to 1. When this is the case, Eq. 3.25 is satisfied.

To achieve global consistency, two series of local computations known as message passes are performed. The first series (evidence collection) involves passing messages in the direction of an arbitrarily chosen clique C, called the root. The second series (evidence distribution) involves passing messages away from C.

Each message pass is defined by two update equations, known as *projection* and *absorption* [129]. These must be applied sequentially, and together form a complete message pass. Projection works from clique towards separator set and is defined thus:

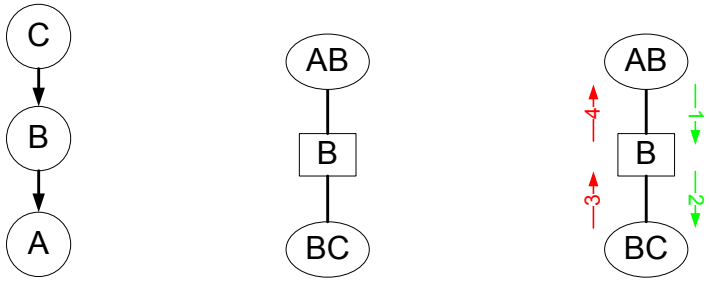
$$\phi^*(S) = \sum_{C \setminus S} \phi(C), \tag{3.28}$$

where S is the separator set and C is the neighbouring clique.

Absorption works from separator set towards clique, and is defined as

$$\phi^*(C) = \phi(C) \frac{\phi^*(S)}{\phi(S)}. \tag{3.29}$$

Example: message passing on a simple chain graph



(a) simple chain graph Bayesian network

(b) junction tree for the Bayesian network

(c) junction tree with update ordering

Figure 3.7 — Exemple of Bayesian network and corresponding junction tree for message passing

To illustrate the message passing procedure, we take as an example the simple junction tree of Fig. 3.7(b) with two cliques, $C_1 = \{A, B\}$ and $C_2 = \{B, C\}$, and a separator set $S = B$ between the two cliques, we initialise the potentials as follows:

$$\begin{aligned} \phi(A,B) &= P(A|B) \\ \phi(B,C) &= P(B|C)P(C) \\ \phi(B) &= 1 \end{aligned}$$

Now, it is easy to see if we choose C_2 as a root, the first message pass consists in a projection followed by an absorption (respectively steps 1 and 2 in Fig. 3.7(c)))

$$\phi^*(S) = \sum_{C_1 \setminus S} \phi(C_1) = \phi^*(B) = \sum_{A} \phi(A,B) = \sum_{A} P(A|B) = 1 \tag{3.30}$$

$$\phi^*(C_2) = \phi(C_2)\frac{\phi^*(S)}{\phi(S)} = \phi^*(B,C) = \phi(B,C)\frac{\phi^*(B)}{\phi(B)} = P(B|C)P(C) = P(B,C) \tag{3.31}$$

The second message pass consists in a projection followed by an absorption (respectively steps 3 and 4 in Fig. 3.7(c)))

$$\phi^{**}(S) = \sum_{C_2 \setminus S} \phi^*(C_2) = \phi^{**}(B) = \sum_{C} \phi(B,C) = \sum_{C} P(B|C)P(C) = P(B) \tag{3.32}$$

$$\phi^*(C_1) = \phi(C_1)\frac{\phi^{**}(S)}{\phi^*(S)} = \phi^*(A,B) = \phi(A,B)\frac{\phi^{**}(B)}{\phi^*(B)} = P(A|B)P(B) = P(A,B) \tag{3.33}$$

To see that the result is correct, we can apply Eq. (3.25)

$$P(V) = \frac{\prod_i \phi_{c_i}}{\prod_j \phi_{s_j}} = \frac{\phi^*(C_1)\phi^*(C_2)}{\phi^{**}(S)} = \frac{P(B,C)P(A,B)}{P(B)} = P(C)P(B|C)P(A|B), \tag{3.34}$$

which indeed correspond to the factorisation of the Bayesian network in Fig. 3.7(a).

Example: message passing on the Asia Bayesian network

We will again use the Asia Bayesian network as an example. Once the Bayesian network $\mathcal{G}$ in Fig. 3.2 has been transformed to the junction tree $\mathcal{G}_j$ shown in Fig. 3.6, the first step is to initialise the potentials in the junction tree, which for the cliques can be done using the conditional probabilities encoded by the Bayesian network $\mathcal{G}$, and for the separator sets by setting them to 1. From the joint distribution encoded by the Asia network given in Eq. (3.16), we can initialise the 6 clique potentials.

$$\begin{aligned}
\phi(E,L,T) &= P(E|T,L) \\
\phi(B,L,S) &= P(S)P(L|S)P(B|S) \\
\phi(D,E,B) &= P(D|E,B) \\
\phi(B,L,E) &= 1 \\
\phi(A,T) &= P(A)P(T|A) \\
\phi(X,E) &= P(X|E)
\end{aligned}$$

The 5 separator sets potentials can also be initialised.

$$\begin{aligned}
\phi(B,L) &= 1 \\
\phi(E,L) &= 1 \\
\phi(B,E) &= 1 \\
\phi(T) &= 1 \\
\phi(E) &= 1
\end{aligned}$$

One possible order or computation for the message passes is shown on Fig. 3.8. After all updates have been performed, the tree will be globally consistent, and the potentials will contain marginal probabilities corresponding to the variables in each cluster and separator set.

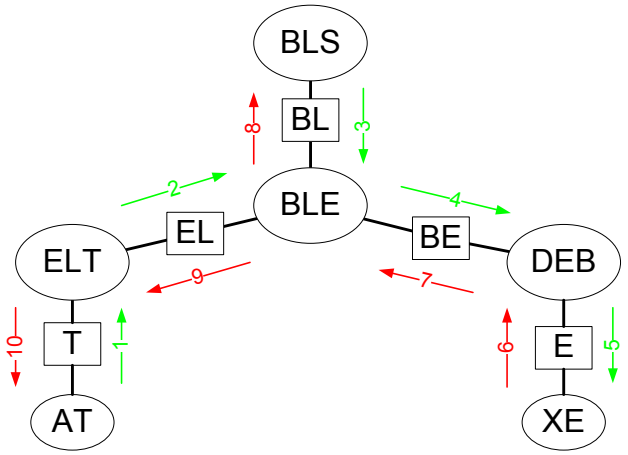


Figure 3.8 — Example order of computation of message passes for the junction tree of the Asia network. Passes 1-5 (in green) are evidence collection, and passes 6-10 (in red) correspond to evidence distribution

3.5 Pattern recognition with Bayesian networks for biometric authentication

3.5.1 Discrete and continuous nodes

Discrete nodes have multiple uses in Bayesian networks for biometric authentication: since biometric verification is a two-class problem, the ground truth (true user identity) is a binary random varable, which we denote Ω. $\Omega = 1$ corresponds to the real-world event "the biometric data belongs to the client", while $\Omega = 0$ corresponds to "the biometric data comes from an impostor". Correspondingly, classifier decisions are also binary random variables, which we denote CID (classified identity). Again, this can be instantiated to correspond to the real-world event "the classifier accepts the identity claim" (encoded as $CID = 1$) or to "the classifier reject the identity claim" ($CID = 0$). In some cases, quality measures (see Chapter 5) can also be discrete. Lastly, nodes indexing mixture components in mixture distributions are discrete.

Continuous nodes, typically representing a Gaussian distribution which can be univariate or multivariate, are used to model continuous random variables such as biometric signals, or more commonly features extracted from biometric signals, quality measures (denoted QM), and classifier output scores (denoted Sc).

For the remainder of this thesis, we adopt the pictorial convention that discrete random variables are represented as rectangular nodes, while continuous random variables are represented as round nodes.

3.5.2 Visible and hidden nodes

Since biometric authentication is cast as a supervised learning problem, nodes corresponding to class labels are observed during training (as explained in Section 3.3.1). However, in testing (inference), the class nodes are hidden, and the most likely value must be computed using one of the inference algorithms presented in Section 3.4.

The model being trained can also have hidden nodes because it is desirable that, for example, mixing weights be trained on data rather than hard-coded (see Section 3.3.1). This is typically the case in hidden Markov modelling with mixture density outputs, or in Gaussian mixture modelling.

Henceforth, visible nodes will be represented as shaded in gray, while hidden nodes will be left white.

3.5.3 Parameter learning and inference

Going back to Section 1.2.1 and Figure 1.1, the *user model creation* and *background model creation* steps are implemented in Bayesian networks by either maximum likelihood learning (Section 3.3.1) or expectation-maximisation (Section 3.3.1). The *matching* step is performed by using one of the inference algorithms presented in Section 3.4.

In some cases, the *Preprocessing* step could also be achieved by using a Bayesian network, for example it is possible to train a model to segment speech and pause portions of speech.

3.6 Summary

In this Chapter we have reviewed and defined basic notions of graph theory and insisted on directed graphical models, of which Bayesian networks, another name for directed acyclic graphical graphs, are a subset. Nodes of the network are used to represent probability distributions, both discrete and

continuous. In Bayesian networks, the independence relationships in the data can be represented by the topology of the network, and a practical criterion, directional separation (d-separation) can be used to verify which independences and conditional independences hold.

If the structure of the network is fixed, for example by a human expert, learning algorihtms can be applied to learn the probability distibutions in each node. The probability distributions in hidden nodes can be trained via the expectation-maximisation algorithm.

Once the parameters of the distributions have been estimated in the whole network, it is possible to use it for inference over variables of interest. One exact algorithm for inference is the junction tree algorithm, which through a series of graphical operations transforms the network into a tree structure. Inference is then equivalent to passing messages back and forth over the tree, until the tree is globally consistent.

Part II

Probabilistic models for multi-classifier biometric authentication with quality measures

Unimodal biometric verification with Bayesian networks

4

4.1 Introduction

The fundamental building block in multiple classifier systems is the unimodal classifier. Many different types of classifiers could be used, but we focus on Bayesian networks as they are very well suited to act as base classifiers, and offer great flexibility in modelling of raw data or features. By modelling user data as probability densities, we take into account the uncertainty and variability inherent in biometric samples. While focusing on signature and speech-based authentication, we show that some modelling principles in Bayesian networks are applicable to a range of biometric authentication modalities.

Furthermore, many of the techniques that are exposed in this chapter and put to use for unimodal verification, such as multivariate modelling and mixture modelling, will be later reused in estimating reliability (Chapter 6), and building multiple classifier systems (Chapter 7, Chapter 8), all within the framework of Bayesian networks.

In Section 4.2, we show two approaches to modelling continuous multi-dimensional data with Bayesian networks. Section 4.3 shows how mixture modelling can be achieved, in particular Gaussian mixture modelling. Section 4.4 shows an application of the Bayesian network equivalent of a Gaussian mixture model to speaker verification, and Section 4.5 shows more specifically the processing tasks needed to apply the same topology to signature verification, offering a comparison with state-of-the-art models for signature verification.

4.2 Bayesian network modelling of multi-dimensional data

Much of the data used in pattern recognition for biometric authentication is in form of real-valued feature vectors. There is therefore a need to model multivariate, continuous random variable distributions. Bayesian networks can handle multi-dimensional data without special problems, since the theoretical framework does not need to be modified to accomodate this kind of data. However, the implementation needs to be carefully considered to avoid numerical problems [199].

Two main possibilities exist to model continuous multivariate data in Bayesian networks: the scalar approach, and the vector approach.

4.2.1 Scalar approach

In the scalar approach, each feature vector component is considered as a separate random variable and assigned to a node. This is the method used by naïve Bayes and tree-augmented naïve Bayes (TAN) classifiers (see Section 7.2). The dependence between feature vector components has to be encoded explicitely in the network topology, adding or removing arcs as needed. Each Gaussian node has a mean and a variance parameter, and possibly a regression vector if it has continuous parents. The regression vector quantifies the strength of the influence of the parent node's value on the child node. The advantage of this approach is that strong modelling constraints can be built into the graph, avoiding overfitting. The disadvantage is that modelling assumptions may be unsupported by data.

A continuous Gaussian node x_i with continuous Gaussian parents $pa(x_i) = x_1, \ldots, x_{i-1}$ has the following conditional density function:

$$P(x_i|pa(x_i)) = \frac{e^{-\frac{1}{2}(\frac{x_i-u_i}{\sigma_i^2})}}{\sqrt{2\pi\sigma_i^2}}, \tag{4.1}$$

where the u_i term represents the influence of the parents:

$$u_i = \mu_i + \sum_{j \in pa(x_i)} b_{ji}(x_j - \mu_j), \tag{4.2}$$

where the b_{ji} are regression coefficients associated with the arcs between parents and children [27, 199].

4.2.2 Vector approach

The second approach to modelling continuous multivariate data is to use vector-valued nodes. In this case, each multivariate Gaussian node has a mean vector and a covariance matrix. the correlations between feature vector components will be learned on a training set. It is of course possible to arbitrarily change elements in the covariance matrix to reflect modelling assumptions, as in the covariance selection approach [66]. If the training set is sufficiently large and is a good match for the test set, the "vector-valued node" approach should result in good covariance estimates and learn valid relationships between feature vector components.

For base classifiers, we tend to favour the vector approach, as it is more compact and covariance matrices (rather than regression matrices) are commonly used in literature. For other applications such as multiple classifier fusion, we use either the scalar approach or the vector approach. Equation (7.29) shows an example of the standard form of the covariance matrix for a vector approach to a two-classifiers fusion problem.

4.2.3 Equivalence of the approaches

Shachter and Kenley [285] have shown that there is a strict equivalence between the multivariate normal distribution and a fully connected Bayesian network with scalar Gaussian nodes, under condition that the regression weights and variances be set appropriately.

Let us define matrix $\mathbf{D}$ as the diagonal matrix containing variances of each random variable, and matrix $\mathbf{B}$ as the regression matrix holding the regression weight vectors over the random variables. The columns of $\mathbf{B}$ represent the regression weight attached to edges from the parent variables*. Further defining $\mathbf{U} = (\mathbf{I} - \mathbf{B})^{-1}$, the covariance matrix of a multivariate distribution can indeed be factored as

$$\mathbf{\Sigma} = \mathbf{U}'\mathbf{D}\mathbf{U} \tag{4.3}$$

It should be noted that the link between Bayesian network reprentation and other models for correlation of multivariate data can also be established via structural equation models [30] or covariance structure models [41], two approaches we do not pursue here.

4.3 Gaussian mixture modelling with Bayesian networks

The Gaussian mixture model (GMM), also called mixture of Gaussians, is a very flexible model that can be used to approximate the shape of any probability distribution if enough mixture components are used [190].

In biometric authentication, GMMs are a very common model and have been applied to a large number of modalities. They are the dominant model in text-independent speaker verification. The GMM model we proposed for signature verification (see Section 4.5) has been used with good results [251, 280, 298], and it also has been applied to writer identification [183].

Since the implementation of Gaussian mixture models without resorting to Bayesian networks is more common, but theoretically equivalent, we first adopt a notation for GMMs without reference to Bayesian network terminology. With a D-dimensional feature vector $\mathbf{o}_t$ part of a complete observation sequence $\mathbf{O} = [\mathbf{o}_1, \ldots, \mathbf{o}_T]$, the general form of a probability density for an M-Gaussian pdf components GMM $\mathbf{\Theta}_M$ is:

$$p(\mathbf{o}_t; \mathbf{\Theta}_M) = \sum_{m=1}^{M} c_m \frac{e^{-\frac{1}{2}(\mathbf{o}_t - \boldsymbol{\mu}_m)' \mathbf{\Sigma}_m^{-1} (\mathbf{o}_t - \boldsymbol{\mu}_m)}}{|\mathbf{\Sigma}_m|^{\frac{1}{2}} (2\pi)^{\frac{D}{2}}}. \tag{4.4}$$

Where c_m is the Gaussian component weight (mixing coefficient) with the constraints that

$$\sum_{m=1}^{M} c_m = 1 \quad \text{and} \quad c_m \geq 0, \tag{4.5}$$

μ_m is the component mean vector, and $\mathbf{\Sigma}_m$ is the component's covariance matrix. If the elements in the feature vector are uncorrelated (or assumed to be), the covariance matrix becomes diagonal and Equation (4.4) can be simplified to:

$$p(\mathbf{o}_t; \mathbf{\Theta}_M) = \sum_{m=1}^{M} c_m \prod_{d=1}^{D} \frac{e^{-\frac{1}{2}\frac{(o_{td} - \mu_{md})^2}{\sigma_{md}^2}}}{\sqrt{2\pi\sigma_{md}^2}}. \tag{4.6}$$

*A regression weight of 0 from a node to another is equivalent to having no connection in the DAG [120]

Using diagonal covariance matrices reduces the number of free parameters $N(\mathbf{\Theta})$ in the model, from

$$N(\mathbf{\Theta}) = (M-1) + M\big(D + D(D+1)/2\big) \tag{4.7}$$

to

$$N(\mathbf{\Theta}) = M - 1 + 2MD. \tag{4.8}$$

Also, diagonal covariance matrices reduce the number of operations for likelihood computations. However, if some degree of correlation exists between the features, as is often the case, the number of Gaussian components will need to be increased to account for it.

A Bayesian network that represents a class-conditional GMM, which we call a BN/GMM model, is shown in Fig. 4.1.

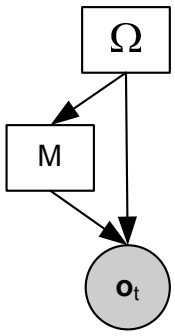


Figure 4.1 — Bayesian network representation of a GMM

In Fig. 4.1, Ω is the class variable, M is the discrete random variable whose probability $P(M|\Omega)$ represents the mixing coefficient, and $\mathbf{o}_t$ is an observation vector. The number of Gaussian components in the mixture is determined by the number of possible states of M. The joint probability distribution over these random variables is factored as

$$P(\Omega, M, \mathbf{o}_t) = P(\Omega)P(M|\Omega)P(\mathbf{o}_t|M,\Omega), \tag{4.9}$$

where $P(\Omega)$ is the class prior, $P(M|\Omega)$ is an $M \times |\Omega|$ table corresponding to the class-conditional mixing coefficients, and the $P(\mathbf{o}_t|M,\Omega)$ term is a set of multivariate Gaussian distributions with mean μ_m and covariance $\mathbf{\Sigma}_m$, indexed on the class and the mixing coefficient. Effectively, this last term is modelled as a collection of M Gaussian distributions for each class. Taking the marginal on the observation vector yields

$$P(\mathbf{o}_t) = \sum_{\omega\in\Omega} P(\omega) \underbrace{\sum_{M} P(M|\omega)P(\mathbf{o}_t|M,\omega)}_{\text{weighted sum of Gaussians}}. \tag{4.10}$$

This model is trained by providing class labels along with observation vectors. The mixing weights remain hidden, and are learned by the EM algorithm (see Section 3.3.1). It can then be used for inference using message passing on its junction tree, with the Ω and M nodes hidden and the $\mathbf{o}_t$ node visible.

If we assume that the distribution of the features independently of time is discriminative, this model can be used to learn multidimensional time-series data. Indeed, we have applied it to signature data with results at least as good as those obtained with hidden Markov models (see Sections 4.5

and 4.4 respectively). We attribute this to the fact that clustering all the data into one state allows for better density estimates than segmenting it into states.

4.3.1 2-class BN/GMM models: the posterior approach

Remembering that biometric verification for a set of U users is not a U-classes problem, but a set of U 2-class problems, the first approach for using the BN/GMM model in biometric authentication is to train U 2-class models. The first class ("client") is trained with labelled data from each user, while the parameters of the second class ("impostor") are estimated using training data from other users. The topology used in this case is that presented in Figure 4.1, and the class node has a cardinality of two.

Computing verification scores using posteriors

The quantity of interest, $P(\Omega|\mathbf{O})$ is computed by first running a full message pass on the observation sequence $\mathbf{o}_t$ (see Section 3.4.2), which has the effect of making the junction tree globally consistent, then by obtaining the marginal probability for the class Ω.

In the case of the model of Figure 4.1, the set of cliques identified after moralisation and triangulation is a singleton $\mathcal{C} = \mathcal{C}_1 = \{\Omega, M, \mathbf{o}_t\}$. The potential $\phi(\mathcal{C}_1)$ is initialised to the corresponding term in the factorisation of the joint probability, that is

$$\phi(\mathcal{C}_1) = P(\Omega)P(M|\Omega)P(\mathbf{o}_t|M,\Omega) \tag{4.11}$$

Since this Bayesian network has only one clique, message passing (evidence collection and evidence distribution) has no effect, except to formally guarantee global consistency. Thus, the clique potential is equivalent to the result of computing the probabilities on the factored joint distribution.

Once the clique potential is obtained, it is necessary to compute the marginal probability for the class variable Ω. This is done by summing over the values of the M node.

The decision rule is then the Maximum A Posteriori (MAP) decision rule, which is equivalent to the Bayesian decision rule of Equation (1.2).

4.3.2 1-class BN/GMM models: the likelihood approach

The second approach for using the BN/GMM model in biometric authentication is to train U 1-class user models and between 1 and U background models. The 1-class user models $\mathbf{\Theta}^u$ are trained on a subset of data containing only feature vectors from user u, and the background models are either trained on the pooled training data of all users*, resulting in using the same background model for all users, or on U different sets of "cohort" users differing from user u, resulting in U user-specific background models†.

The appropriate topology for this approach is to choose a cardinality of 1 for the class node for each of the U user models, as well as for the background model(s). Furthermore, the class prior $P(\Omega)$ is set to 1. Another possibility is to remove the class node entirely, resulting in the topology shown in Figure 4.2

This method offers two main advantages over the posterior approach. The first is that, assuming a single background model is used, the training time is lower because user models are trained on far fewer training vectors – in the posterior approach, each user model contains a copy of the world

*the background users can also come from another database altogether

†This is a common approach in speaker verification, see for instance [268].

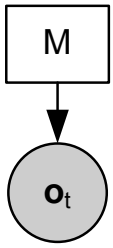


Figure 4.2 — Bayesian network representation of a GMM for the likelihood approach

model. The second is that we can easily choose a different cardinality for the hidden node in the world model than in the user model, in order to speed up training and inference.

Computing verification scores using likelihood extracted from potentials

In the likelihood approach, keeping with well-established practice in speaker verification, we are interested in the logarithm of the ratio of the likelihood of the data given the user model to the likelihood of the data given the background model, as opposed to the posterior probability of the access belonging to a client. More precisely, a verification score is obtained as a ratio of the likelihood that observation $\mathbf{O}$ is seen given the model for user u to the likelihood that any other user produced test presentation $\mathbf{O}$. In other words, the score $S(\mathbf{O}, \mathbf{\Theta}^u, \mathbf{\Theta}^-)$ shows how different the test presentation is from any other presentation in the world model. It is computed as follows:

$$S(\mathbf{O}, \mathbf{\Theta}^u, \mathbf{\Theta}^-) = \log p(\mathbf{O}; \mathbf{\Theta}^u) - \log p(\mathbf{O}; \mathbf{\Theta}^-) \tag{4.12}$$

The log-likelihood of a feature vector with respect to a model can be elicited by running the junction tree algorithm (see Section 3.4.2) over the Bayesian network.

In the case of the model of Figure 4.2, the set of cliques identified after moralisation and triangulation is a singleton $\mathcal{C} = \mathcal{C}_1 = \{M, \mathbf{o}_t\}$. The potential $\mathcal{C}_1$ is initialised to the corresponding term in the factorisation of the joint probability, that is

$$\phi(\mathcal{C}_1) = P(M)P(\mathbf{o}_t|M) \tag{4.13}$$

Since this Bayesian network has only one clique, message passing (evidence collection and evidence distribution) has no effect, except to formally guarantee global consistency. The likelihood of each Gaussian component (scaled by the mixing coefficient $P(M|\Omega)$) is found by inspecting the corresponding clique potential.

The decision rule is then the maximum likelihood decision rule.

4.4 Speaker Verification with Bayesian networks

4.4.1 Introdution

In this section we reformulate a state-of-the-art approach for single-classifier speaker verification (Gaussian mixture modelling with universal background models [247]) in terms of the corresponding Bayesian network, insisting on issues specific to speaker verification.

Later, in Section 4.5, we show how a similar Bayesian network topology can be applied to signature verification, and highlight the differences.

4.4.2 Preprocessing

The first operation performed on the speech signal is to remove the mean of the time-domain signal to counter DC bias which may have been introduced by the analog-to-digital conversion process. Then, the speech is passed through a voice activity detector which removes the silent parts of the signal (two algorithms for doing so are described in Section 5.5.1).

4.4.3 Features

The purpose of extracting acoustic feature vectors from the speech signal is to obtain speaker-dependent information. In the source-filter model of speech production, the speech signal is decomposed into excitation (e.g. vibration of the vocal folds) and filter (corresponding to the shape of the vocal tract.). Since the excitation signal is less discriminative than vocal tract shape*, most acoustic features used for speaker verification concentrate on capturing this latter aspect of the signal.

Popular features include linear prediction cepstral coefficients (LPCC) [96], which are based on a predictive model of speech, Mel-frequency cepstral coefficients [40], which are based on a smoothed and transformed speech spectrum, and have proved successful for both speech recognition and speaker recognition, and PLP coefficients, which are motivated by psychoacoustics [123].

The dimension of a feature vector used for speaker verification is typically around 30. In most speaker verification applications, features are modelled using diagonal covariance matrices.

4.4.4 Model topology, background modelling, and model adaptation

In order to deal with data scarcity, one of the best performing techniques is to train a background model (Universal Background Model or UBM) with a large amount of data, and to then adapt the model to each user given user-specific data. This process is known as MAP (Maximum A Posteriori) adaptation [103, 247], and the resulting system as UBM-GMM.

Since Bayesian networks equivalent to Gaussian mixture models can be used for both the background model and the user models, classical adaptation techniques [247] can be used directly with the model of Section 4.3.2. For each mixture component m, the three parameters that are adapted from the world model are the weight c_m, the mean $\boldsymbol{\mu}_m$, and the covariance matrix $\boldsymbol{\Sigma}_m$. The adaptation equation for the weights is:

$$\hat{c}_m = \left[\alpha_m^c \frac{n_m}{T} + (1 - \alpha_m^c) c_m\right] \gamma, \tag{4.14}$$

where α_m^c controls the amount to which user-specific data is taken into account, n_m is the total responsability of this component (amount of data that is probabilistically assigned to this Gaussian component) given the training data for this user, T is the number of training vectors for this user, and γ is a normalising constant to enforce Equation (4.5).

The adaptation equation for the mean vector is:

$$\hat{\boldsymbol{\mu}}_m = \alpha_m^\mu E_m(\mathbf{O}) + (1 - \alpha_m^\mu) \boldsymbol{\mu}_m, \tag{4.15}$$

where $E_m(\mathbf{O})$ denotes the expectation taken over the training data for this user and this mixture component, again resorting to the probabilistic assignment $P(m|\mathbf{O})$, and α_m^μ controls the amount of adaptation performed.

*Excitation-derived information such as pitch information (which can help distinguish between female and male users, as female users have on average double the pitch of male users) are generally used in addition to a base system modelling features representing the vocal tract shape)

Lastly, the adaptation equation for the covariance matrix is:

$$\hat{\mathbf{\Sigma}}_m = \alpha_m^{\Sigma} E_m(\mathbf{O}^2) + (1 - \alpha_m^{\Sigma})(\mathbf{\Sigma}_m + \boldsymbol{\mu}_m^2) - \hat{\boldsymbol{\mu}}_m^2. \tag{4.16}$$

The adaptation control parameter $\alpha_m^{\cdot}$ controls how much new data should be available before overcoming the prior represented by the world model parameters:

$$\alpha_m^{\cdot} = \frac{n_m}{n_m + r^{\cdot}}, \tag{4.17}$$

where n_m is the probability of a feature vector being the responsibility of component m, $r^{\cdot}$ is called the relevance factor.

In our experiments, we use means-only adaptation, as it has been widely reported to give performance at least equivalent to full MAP adaptation.

4.4.5 Speaker verification experiments and results

Experimental setup

We use the 52-users english subset of the BANCA database and the 295-users XM2VTS database for these experiments. A detailed description of these databases is provided in Section A.1. For BANCA, we first train on G1 and test on G2, then train on G2 and test on G1. For XM2VTS, we train on the evaluation set and test on the test set, according to the Lausanne Protocol Configuration 1.

The preprocessing and feature extraction is the same for both databases: a VAD is run to remove silence portions of the input speech signal before feature extraction, and the features used are 12 MFCCs (no energy) with delta and acceleration coefficients, and cepstral mean normalisation.

A world model is trained from the pooled clean training data of all clients, using 200 diagonal covariance-matrix Gaussian components. Each client's model is then adapted (means only) with their own recordings using MAP adaptation. Instead of using a Bayesian network implementation for the UBM-GMM model, given the size of the datasets, we resort to the equivalent but faster Alize library for speaker recognition [31].

Sensitivity analysis

In this series of experiments we change the most significant tunable parameter in a GMM classifier: the number of Gaussian components. For all experiments, we use diagonal covariance matrices. Figure 4.3 summarises the results. IT can be seen that, for all numbers of mixture components, results on XM2VTS are significantly better. This is a reflection of the data quality: XM2VTS is recorded in noise-free conditions, while the environment in BANCA is much more challenging. For XM2VTS, the performance is very stable from the point were 100 mixture components are used until at least 350 components, while the optimal for BANCA seems to be around 250 Gaussian components. Figure Figure 4.4 shows the DET curves for the case where 250 mixture components are used.

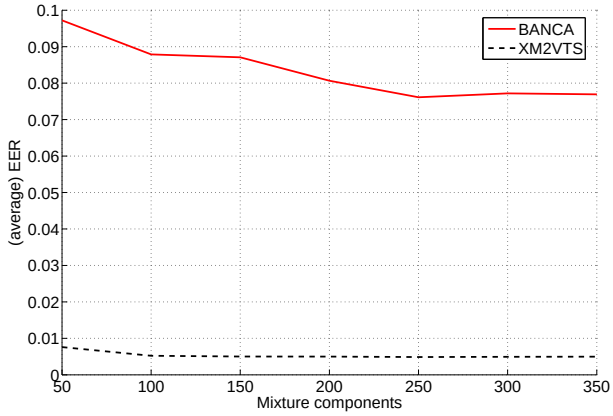


Figure 4.3 — Summary of sensitivity to number of mixture components in two speech databases. Note that BANCA results are an average over G1 and G2, while the XM2VTS results are provided for the test set.

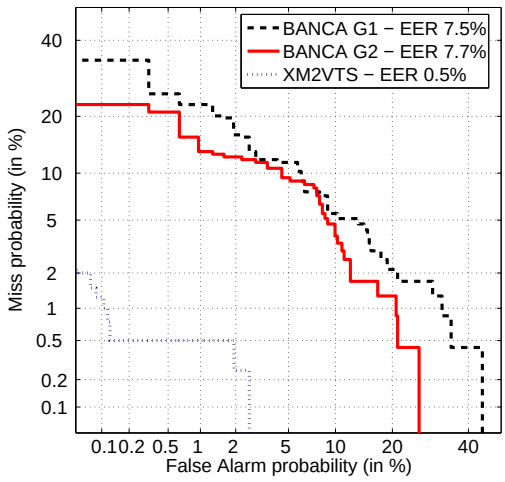


Figure 4.4 — DET curves for speaker verification on BANCA and XM2VTS.

4.5 Signature Verification with Bayesian networks

4.5.1 Introdution

From a signal processing point of view, online* handwritten signatures can be seen as realisations of a multidimensional random process. Once the signal has been preprocessed (Section 4.5.2) and parameterised (Section 4.5.3) properly, many of the analysis and modelling methods used in speaker verification can be applied.

In this section, we present signal processing operations and modelling concerns (Section 4.5.4) that are specific to signature verification, emphasizing the differences with speaker verification whenever possible. Since the state of the art in single-classifier signature verification is generally achieved using hidden Markov models, we expand on the commonalities and differences between the HMM and the GMM approaches (Section 4.5.5).

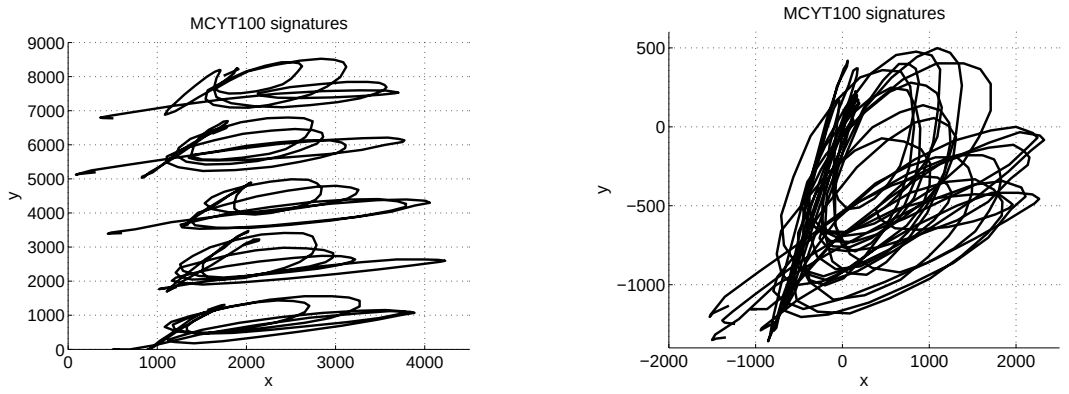
4.5.2 Geometrical preprocessing

To handle intra-user variability, it is necessary to apply some geometrical operations to the raw signature data. Their application depends on the specifics of the database, but we present here the most frequently used. We omit scaling transformations [94] as we have generally found them to be detrimental to probabilistic modelling.

Translation invariance with initial point alignment

Because of differences in acquisition methodologies and inherent variability in initial pen-down position, it is necessary to make the x and y values translation-independent. This is achieved by subtracting the initial x and y values from all subsequent sample points.

Figure 4.5 shows signature data before and after translation invariance transformation.



(a) 5 signatures without translation invariance

(b) 5 signatures with translation invariance

Figure 4.5 — Signature preprocessing: translation invariance by initial point alignment.

Rotation invariance: the eigenvectors approach

In some situations, such as handheld device-based acquisition, it is likely that the orientation of the signature with respect to the horizontal axis of the acquisition surface can be very variable.

*The pen trajectory and dynamics (pressure, azimuth, elevation) are recorded during the signing, as opposed to *offline* signatures where only the 2D trace left on paper is available.

In this case, we estimate the principal axis of the signature by computing the eigenvectors $\mathbf{v}_1, \mathbf{v}_2$ and eigenvalues λ_1, λ_2 of the (x, y) covariance matrix $\mathbf{\Sigma_{xy}}$. The maximal eigenvalue λ_{max} indicates the axis of greatest variance, and the arc tangent of the signature angle can be recovered from the components of the associated eigenvector $\mathbf{v}_{max}$:

$$\theta^{-1} = arctan\Big(\frac{\mathbf{v}_{max_2}}{\mathbf{v}_{max_1}}\Big). \tag{4.18}$$

Then, the $(\mathbf{x}\ \mathbf{y})$ matrix is multiplied by the following rotation matrix:

$$\begin{pmatrix} cos(-\theta^{-1}) & -sin(-\theta^{-1}) \\ sin(-\theta^{-1}) & cos(-\theta^{-1}). \end{pmatrix} \tag{4.19}$$

One advantage of this approach is that it applies equally to western (mostly horizontal) and chinese-like (mostly vertical) signature styles, as it will normalise with respect to the axis of greater variance. Figure 4.6 shows an example of rotation normalisation on the BMEC 2007 mobile signature database.

Again, it should be emphasized that the geometrical preprocessing operations need to be applied only in certain cases. For instance, rotation and scale differences are limited in the MCYT-100 corpus, because the strict acquisition grid, consisting of 3.3 cm x 1.2 cm boxes, constrains both signature orientation and size.

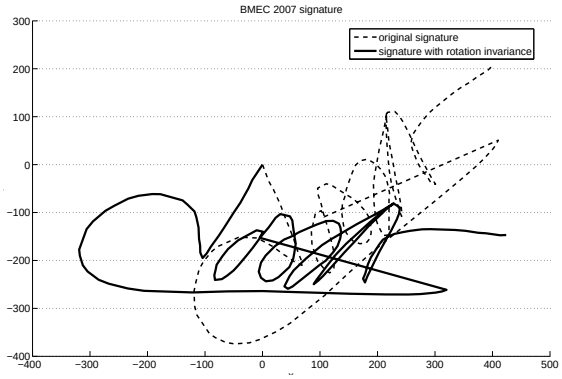
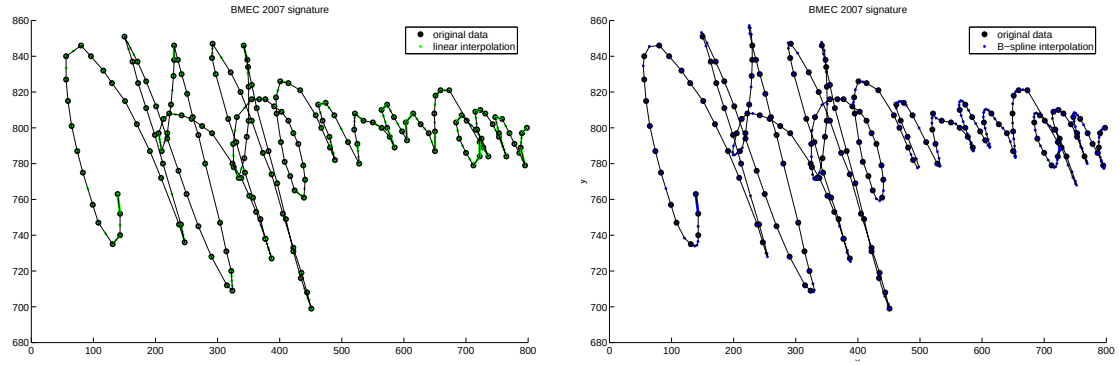


Figure 4.6 — Rotation invariance on the BMEC 2007 database

Handling missing data

On some acquisition platforms lacking a powerful CPU and/or a real-time operating system, such as personal digital assistants, the sensor may intermittently miss acquiring some data due to the scheduler. Missing data is also a frequent occurence in slow and fast strokes. In this case, an effective approach is to interpolate the missing data. We use three algorithms, one which performs linear interpolation over the missing points, one which performs linear interpolation over the missing points even when the pen is lifted up (to emulate the functionality of Wacom-type pen tablets), and another which performs B-spline interpolation [307]. The B-spline model is better motivated physiologically because in first approximation the arm/hand system is a mechanical object with inertia, which cannot produce instantaneous changes of direction. Figure 4.7 shows an example of signature with missing data and two interpolation schemes.



(a) Linear interpolation on pen-down points only

(b) B-spline interpolation

Figure 4.7 — Signature preprocessing for recovery of missing data on BMEC 2007

4.5.3 Features

Features are extracted from the pre-processed signature, which can then be interpreted according to two broad paradigms [230]: function-oriented or parametric. In the function-oriented paradigm (used e.g. in [216]), signals extracted from signature data (such as pressure or velocities) are considered as functions of time, the values of which directly constitute the feature vectors. In the parametric paradigm, local, segmental, or global parameters are computed from the measured signals and used as features.

Local features are extracted at the same rate as the incoming signal: that is, each input sample data vector corresponds to a local feature vector. Examples of local features are instantaneous pressure, radius of curvature, and others. A list of commonly-used local features can be found in [259].

Segmental features are extracted once the signature has been cut into segments. The segmentation paradigms vary, but a segment typically consists of a sequence of points for which some definition of coherence holds. For instance, a signature can be segmented between points of minimal radius of curvature, maximum speed, or zero pressure. More sophisticated segmentation methods can also be used, relying for instance on motor control models [37].

Global features summarise some property of the complete observed signature; for instance the total signing time, pen-up to pen-down ratio, bounding box aspect ratio, etc. A list of commonly-used global features and an algorithm to perform feature selection can be found in [143, 146]

Local features

Signature verification differs from many other biometric modalities because raw digitised data from the sensor can be used directly as features. In addition to this raw data, namely values for the horizontal (x) position, vertical (y) position, pressure (p), azimuth, and elevation of the signing pen, secondary features can be extracted. Two features which typically perform well across a range of databases are the trajectory tangent angle θ_t and the velocitiy v_t, which are computed as

$$\theta_t = \arctan \frac{\dot{y}_t}{\dot{x}_t}, \quad v_t = \sqrt{\dot{x}_t^2 + \dot{y}_t^2}, \tag{4.20}$$

where $\dot{y}_t$, $\dot{x}_t$ indicate first derivatives of y_t and x_t with respect to time. This results in a basis feature vector for each sample:

$$\tilde{\mathbf{o}}_t = [\ x_t, \quad y_t, \quad p_t, \quad \theta_t, \quad v_t\ ]' \tag{4.21}$$

Numerous other features can be extracted from the raw data. We provide here a short summary.

1: horizontal position x_t	2: vertical position y_t	3: normal pressure p_t
4: path tangent angle θ_t	5: total velocity v_t	6: x velocity v_x
7: y velocity v_y	8: total acceleration a	9: x acceleration a_x
10: y acceleration a_y	11: log radius of curvature	12: pen azimuth ϕ_t
13: pen elevation λ_t	14-26: Δ(features 1-13)	27-39: Δ(features 14-26)

Table 4.1 — Frequently used local features for signature verification

Global features

Global features can be useful even though they give inferior performance compared to local features. They can be used in signature classifier ensembles as a way to increase classifier diversity.

While more than 150 global features can be encountered in the literature, Table 4.2 presents only the most frequently used.

1. number of samples (T)	2. signature height (H)	3. signature width (W)
4. H to W ratio	5. T to W ratio	6. avg. velocity ($\bar{v}$)
7. max velocity v_{max}	8. avg. velocity $\div$ max velocity	9. avg. x velocity
10. var. of x velocity	11. n. pts. with +ve x velocity ($N_{v_x>0}$)	12. RMS velocity
13. var. of velocity	14. pen down samples (T_d)	15. time of max velocity $\div T_d$
16. time of max x velocity $\div T_d$	17. RMS acceleration	18. avg. acceleration
19. var. of acceleration	20. avg. pressure ($\bar{p}$)	21. max pressure (p_{max})
22. point of max pressure ($t_{p_{max}}$)	23. avg. azimuth	24. avg. elevation ($\bar{\lambda}$)
25. avg. y velocity	26. x y velocity correlation	27. first moment
28. max pressure-min pressure	29. max x velocity	30. avg. x acceleration
31. max y velocity	32. avg. y acceleration	33. var. of pressure (σ_p)
34. point max. velocity $\div T_d$	35. num. points with negative x or y velocity $\div T_d$	
36. max. acceleration	37. num. points with positive x or y velocity $\div T_d$	
38-46. tangent histogram in 8 quadrants: $S_q = card\left\{\theta_t : (q-1)\frac{\pi}{8} < \theta_t < q\frac{\pi}{8}\right\} \div (T-1)$ where $t = 2, \ldots, T$ and $q = 1, \ldots, 8$		

Table 4.2 — Frequently used global features for signature verification

A combination of 12 features that has proved effective over very diverse signature databases is

$$\tilde{\mathbf{o}} = [\ T, \quad \bar{v}, \quad \frac{\bar{v}}{v_{max}}, \quad N_{v_x>0}, \quad T_d, \quad \bar{p}, \quad \sigma_p, \quad p_{max}, \quad t_{p_{max}}, \quad \bar{\lambda}, \quad S_1 \quad , \frac{N_{v_x>0}}{T_d}\ ]'. \tag{4.22}$$

For simple sensors that do not provide pressure or angle values (such as mobile devices and some signature tablets), the pressure-related features ($\bar{p}, \sigma_p, p_{max}, t_{p_{max}}$) and elevation-related features ($\bar{\lambda}$) are removed .

Delta features

The first and second time derivative of the features, referred to in speech processing as delta features, is also known to be useful in signature verification [216, 217].

Since the signal is discrete, we use a numerical approximation of a first order derivative. By definition:

$$df(\cdot) = lim_{\epsilon \to 0} \frac{f(\cdot + \epsilon) - f(\cdot)}{\epsilon}, \tag{4.23}$$

which we replace by second order regression using the central difference approximation. for the tth term in the vector sequence of length T:

$$df(\cdot_t) \approx \frac{f(\cdot_{t+1}) - f(\cdot_{t-1})}{2} + \frac{f(\cdot_{t+2}) - f(\cdot_{t-2})}{4} \tag{4.24}$$

Feature postprocessing

Because the dynamic ranges of the different features vary widely, each individual feature $\tilde{o}_{dt}$ where $d = 1, \ldots, D$ is standardised to a zero-mean, unit variance distribution using:

$$o_{dt} = \frac{\tilde{o}_{dt} - \mu_{\tilde{o}_d}}{\sigma_{\tilde{o}_d}} \tag{4.25}$$

This operation is equivalent to cepstral mean subtraction and reduction [26], a common technique in speaker verification. In signature verification, it is generally known to give better verification results in hidden Markov modelling [216], and is also common practice in other areas of pattern recognition [302].

The difference is empirically visible in probabilistic modelling: by keeping all values in a covariance matrix to approximately the same ranges, precision issues are avoided in numeric computations (such as inversion or optimisation).

4.5.4 Bayesian networks for signature verification

Once the features are extracted, the same Bayesian network model is used for signature verification as for speaker verification, with a few important differences. First, the topology of the model is not exactly the same - more exactly, hidden variables do not have the same dimensions. Second, the background modelling is used for different reasons. Third, the training procedure is not based on MAP-adaptation from a background model.

We now look at these differences in more details.

Choosing the cardinality of the hidden mixture node

An important design aspect is the choice of the number of discrete states the hidden node can assume, which corresponds to the number of Gaussian components the mixture model contains.

Based on the system performance in terms of error rate for a chosen set of features over several signature databases, it was found that the optimal number of Gaussian components in the mixture is significantly smaller than that used in speaker verification tasks, where background models are often trained using between 200 and 1000 components with diagonal covariance matrices [26]. This is imputable to three inter-related facts: first, speaker verification systems typically use signal transformations which only approximately decorrelate the speech features, whereas most of the times in practical systems the combination of signature features used in signature verification are correlated much more weakly. Second, the size of the feature vectors used in signature verification

is typically smaller (15 dimensions being about a maximum for most methods in the literature) than what is commonly used in speaker verification (where 30 dimensions are not uncommon). Third, the amount of training data per user is much less in signature verification than in speaker verification: assuming 5 training signatures sampled at 100 Hz with an average duration of between 1 and 2 seconds from which local features are extracted, between 500 and 1000 training vectors are available. In speaker verification, assuming between 30 seconds and 1 minute of training data (a low amount for some tasks) with a frame rate of 10 ms, between 3000 and 6000 training vectors are available.

As with any pattern recognition system, the model capacity should neither be so low that the likelihood of observing the training data is very small, neither be so high that the model looses the capacity to predict unseen values of the modelled random variables. In practical cases, the likelihood of the training data given the model increases with the number of model parameters, so the likelihood function in itself is not a good indicator of overfitting. In [10] it has been suggested to use the knee in the curve of the *increase* of the log-likelihood to determine the optimal number of mixtures. However, this approach does not explicitly penalise more complex models.

The Minimum Description Length (MDL) principle [264] can be used to obtain a cost function that balances modelling errors and model complexity. Given a set of trained model parameters for different model orders, the number of training samples and the number of free parameters in the model, the minimum of the MDL cost function indicates the model that can represent both the data and the model parameters in the most compact fashion. For an M-components mixture model, an approximate expression for the MDL cost function can be written as

$$\mathrm{MDL}(\boldsymbol{\Theta}_M, M) = -\log p(\mathbf{O}; \boldsymbol{\Theta}_M) + \frac{1}{2} N(M) \log T \tag{4.26}$$

The first term is small if the model fits the data well, thus reducing modelling errors. The second term is large if the model has a large number of parameters, thus penalising complex models. The MDL criterion has been shown to be a consistent estimator of GMM order for a variety of problems [265].

As an experiment, we trained four different full-covariance matrix GMMs with 8, 16, 24 and 32 Gaussian mixture components respectively. This showed that $\boldsymbol{\Theta}_{16}$ had slightly higher minimum description length than $\boldsymbol{\Theta}_{32}$, but that both 8- and 16-components models had significantly higher MDL values. The results are shown in Figure 4.8. This suggests that using either 16 or 32 full-covariance components per model will result in the most appropriate user signature models.

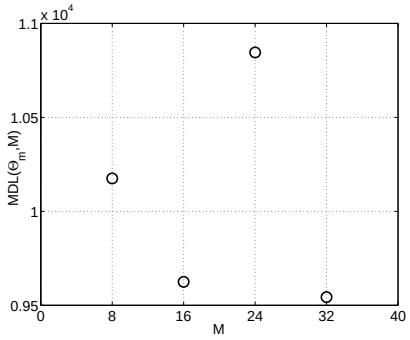


Figure 4.8 — Average MDL values for all users in the MCYT-50 with models using 8, 16, 24 and 32 full-covariance matrix Gaussian components

However, tests on diagonal-covariance matrix GMMs with 16, 32 and 64 components showed

that the MDL criterion tended to under-estimate the optimal number of components; it was not found to be a good predictor of optimal model order for classification performance in the diagonal covariance case. This is in line with the findings in [128, 265]. Also, Friedman et al. [93] have noted that the MDL score is not a good choice for scoring Bayesian network models, unless the amount of data is very large (MDL is asymptotically correct). This criticism is also valid for the Bayesian Information Criterion (BIC) score [58].

Kohavi [159] has reported that for stable classifiers* the variance of the cross-validation estimate of accuracy is itself stable, with only little influence from the number of folds. Additionally, Kearns et al. [145] have shown that within reasonable constraints, cross-validation-based model seletion has tighter error bounds than coding-length based methods such as MDL. Since we use diagonal covariance matrices to reduce the number of parameters in user models, the MDL score is not appropriate, as mentioned above. We therefore select the number of Gaussian components by cross-validation on a development set, despite the penalty in computational expense†.

In modelling global features, it is imperative to choose a very low cardinality for the hidden mixture node, as in most experimental protocols less than 10 signatures are provided. This means that very few training vectors (one per signature) will be available.

Background modelling

In signature verification, the role of the background model is not to attenuate channel effects by normalising the score as is one of the goals of using background models in speaker verification, but to provide a set of random impostors against which to discriminate. Given that most "skilled" forgeries available in academic databases are in fact at most vaguely resemblant to the original, this is an effective strategy to adopt. Indeed, we have shown in [146, 259] that by choosing features that maximise the distance between users, the separation with forgers is also increased.

We use the training data of all users as a background model, but cohort approaches are possible [217].

Model adaptation

We do not use model adaptation for training user models in signature verification. Our experiments on adaptation schemes for MCYT-100 (using means-only or full adapation, with different relevance factor settings) have shown that there is no gain to be obtained from adaptation (not reported here). This is presumably because there is not much in common between signatures from different people, and the result is a bad initialisation for the Gaussian mixture components.

4.5.5 Comparing the Bayesian network model and hidden Markov models for signature verification

Hidden Markov models are well suited for the modelling of doubly-stochastic processes, where the behaviour of the features is expected to be time-dependent. At each "time instant", the model instantaneously jumps from a state to another, and observes a feature vector represented by each state's output distribution.

*in the sense that the trained parameters do not vary very much with different subsets of the training sets. Typically, unpruned CART trees are not stable classifiers, while Gaussian mixture models can be considered stable – this is a reason why boosting or bagging Gaussian mixture models typically offers no improvement.

†Note that even for scoring methods such as MDL, models have to be trained for all candidate number of Gaussian mixtures, since these methods typically require the computation of a data likelihood.

Dynamic Bayesian networks [63] can be used to represent hidden Markov models. In the simple case, the first-order Markov assumption means each state s is independent of the parent of its parent given its parent: $s_{t+1} \perp\!\!\!\perp s_{t-1}|s_t$.

The structure of an HMM is described by a number of states S, the matrix of transition probabilities between states $\mathbf{A}$, and the initial probabilities of each state π.

The initial probabilities and the transition probabilities are defined as follows:

$$\begin{aligned} \pi_i &= P(s_i), \\ a_{ij} &= P(x(t) = s_j | x(t-1) = s_i) \quad \text{for} \quad 1 \leq i \leq S, 1 \leq j \leq S, \end{aligned} \tag{4.27}$$

where $x(t)$ is the state at time t and a_{ij} is the probability of making a transition from state i to state j. The initial and transition probabilities have to satisfy the constraints:

$$\begin{aligned} \textstyle\sum_{i=1}^{S} P(s_i) &= 1, \\ \textstyle\sum_{j=1}^{S} P(s_j|s_i) &= 1 \quad \text{for} \quad 1 \leq i \leq S. \end{aligned} \tag{4.28}$$

For GMMs, both the transition matrix and the initial probabilities vector degenerate to scalars:

$$A_{GMM} = 1, \quad \pi_{GMM} = 1. \tag{4.29}$$

Using an HMM implies the assumption that observations are independent given the state. This particular assumption is very likely to be false for signature data, since the feature values, for instance x and y, are changing slowly with respect to the sampling frequency. However, the same assumption is made in speech processing and can be partly compensated for by using feature derivatives (delta values, see Section 4.5.3).

To fully define an HMM, the parameters of each state's output distribution $b_s(\mathbf{o}_t) = P(\mathbf{o}_t|x(t) = s)$ need to be specified. The output distributions can be discrete, semi-continuous (SCHMM [131]) or continuous (CDHMM). In signature verification, continuous distributions are typically used, though the Bayesian network framework is flexible enough to accomodate discrete feature distributions. A mixture of Gaussians can be used to model the output distribution for each state of a hidden Markov model: the component weights, means and covariance matrices for a single Gaussian component need only be made state-specific with a state index s.

As pointed out by Xuan et al. [323], the Expectation-Maximisation algorithm formulae for iterative re-estimation of component weights, means and covariance matrices are very similar in the GMM and HMM case, and the usage of EM for training GMM parameters can be seen as a special case of the more general EM for training HMM parameters.

Hidden Markov model topologies for signature verification

The most obvious difference between a GMM and an HMM is that HMMs can have more than one state. Apart from the choice of output distribution, the most crucial decision in designing HMMs is to define how many states are needed, and what the possible transitions between them are.

Three broad classes of approaches for discovering the optimal HMM topology and number of states for signature verification exist. The first approach is human expert decision (for instance based on error rates) on a particular topology, where both the transition matrix type and the number of states is fixed. For instance a four-states, left-to-right HMM with skips can be thought to be optimal [324], its performance for the task tested and its topology updated given test results.

The second approach involves fixing the transition matrix type but leaving the number of states free. The data is generally aggressively quantised before applying a structure learning algorithm.

Then, discontinuity between quantised feature values can be taken to mean a change of state. This has been used by Muramatsu and Matsumoto [197], where for Japanese signatures pen positions are quantised to 16 directions and the features used are the quantised angles. Imposing a left-to-right topology with no skips, each change of quantised angle in the feature stream creates a new HMM state. Another approach is to make the number of states dependent on the average number of training feature vectors per signature for each user [73].

The third approach leaves both the transition matrix type and the number of states free; a model is learned directly from the observed features. Stolcke and Omohunddro [296] propose a method in which the data is first modelled by a maximum likelihood HMM, which can reproduce the training data exactly, then states are successively merged to obtain a more general HMM using a posterior probability criterion. The reverse approach has been applied in [295], where states are split according to criteria based on goodness-of-fit or the MDL principle. Automated induction of both HMM topology and number of states has to the best of our knowledge not been applied to on-line signature verification problems.

Over the years, researchers have generally reduced the number of states used in HMMs for signature recognition. Thus, while in 1998 and 1999 between 10 and 30 states [143], respectively between 30 and 40 states [43] have been used, more recently in 2002 between 6 and 12 states [95] and from 2003 onwards as little as 2 states [82] have been shown to be effective. From these earlier results, it appears that the time dynamics of signatures is not as important as the distribution of features.

The number of free parameters in an HMM depends on the topology. For a left-to-right topology, the initial probabilities π are fixed, so only the transition matrix has to be estimated and stored in addition to each state's output distribution. Thus, for a left-to-right S-states HMM with no skips using M diagonal covariance matrix Gaussian components to model the distributions in each state, the number of free parameters $N(S, M)$ is:

$$N(S, M) = 2(S - 1) + S(M - 1 + 2MD). \tag{4.30}$$

In order to enable a principled comparison between multi-state (HMM) and single-state (GMM) models for signature verification, it is important to ensure the number of parameters stays in approximately the same range for the models under comparison. Since the model parameters are estimated from the same finite amount of training data, this should isolate the effect of a time-dependent topology (as opposed to model order effects) on verification performance.

4.5.6 Signature verification experiments and results

Experimental setup

We use three databases for these experiments: The MCYT-100 database [218], the training set of the SVC 2004 database [326], and the BMEC2007 development database. These are presented in more details in Section A.1.

For MCYT-100, the results are presented by training with the first 5 signatures, and testing on the remaining data. This is also the case for BMEC2007, according to the competition protocol. For SVC 2004, results are presented following the experimental protocol of the competition: all EERs are averaged over 10 crossvalidation runs, during which 5 signatures out of the first 10 (first session) are randomly selected for training. For all databases, we do not present results for random forgeries as these generally result in much lower error rates and give an overly optimistic view of performance.

Sensitivity analysis

In this series of experiments we change the most significant tunable parameter in the Bayesian network classifier: the cardinality of the hidden mixture variable, corresponding to the number of Gaussian components in a Gaussian mixture model. For all experiments, we use diagonal covariance matrices.

Figure 4.9 shows a summary of sensitivity to the number of mixture components in terms of EER for the three databases. In all cases, the BN/GMM gives stable results for a wide range of parameter settings. (between 30 and 70 for MCYT-100, 30 and 50 for SVC 2004, and between 30 and 60 for BMEC 2007). More detailed results (DET curves) are presented in Figure 4.10 for MCYT-100, in Figure 4.11 for SVC 2004, and in Figure 4.12 for BMEC 2007.

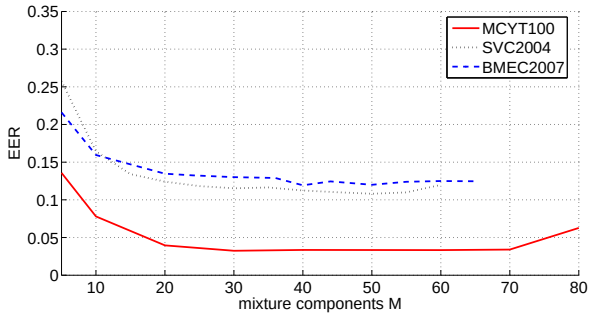


Figure 4.9 — Summary of sensitivity to number of mixture components in three signature databases. Note that SVC2004 results are an average over 10 folds of cross-validation.

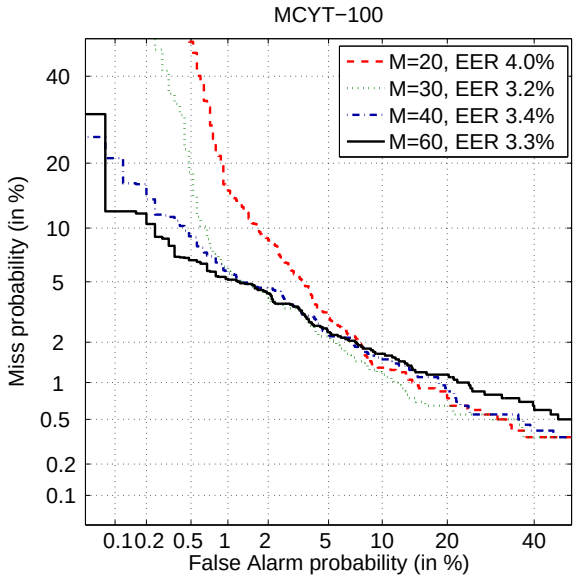


Figure 4.10 — Sensitivity analysis on MCYT-100. The features used are $(x, y, p, \theta, v) + \Delta + \Delta\Delta$.

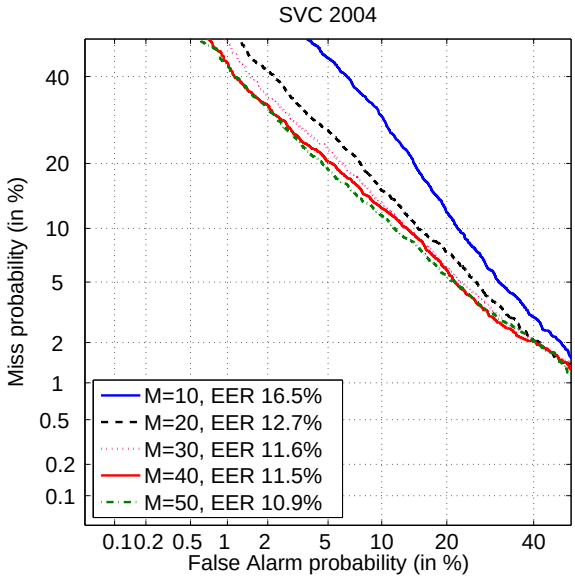


Figure 4.11 — Sensitivity analysis on SVC 2004. The features used are $(x, y, p, \theta, v) + \Delta$. Note that the DET curves are computed using the results produced on all 10 folds, hence their smooth aspect.

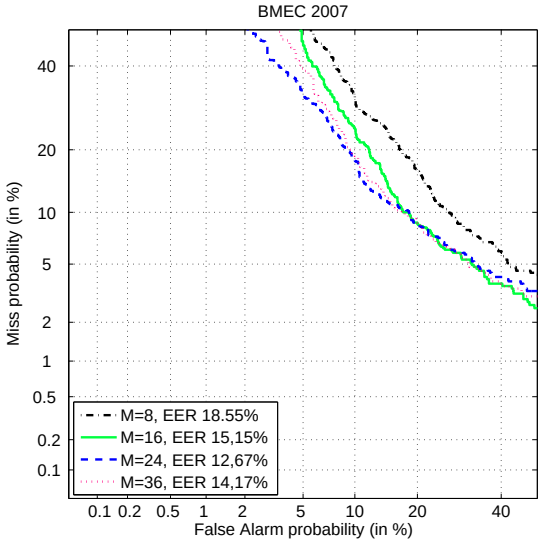


Figure 4.12 — Sensitivity analysis on BMEC 2007. The signal is pre-processed using pen-up interpolation and rotation normalisation. The features used are $(x, y) + \Delta + \Delta\Delta$.

Comparison with Hidden Markov Models

In this series of experiments we compare our Bayesian network classifier with hidden Markov models of equivalent model complexity; that is, comparisons are performed trying to keep the number of free parameters in the same range. In keeping with recent research, it was chosen to compare 5- and 2- states HMMs to the BN/GMM baseline system, using diagonal covariance matrices. The HMMs have a strict left-to-right topology. Both the BN/GMM and the HMM models are initialised using k-means clustering.

In order to assess statistical significance, we compute the EER threshold a posteriori on the classifier outputs, apply it to the output, and get a vector of decisions. We then perform the McNemar significance test [286] with $p = 0.05$.

On MCYT-100, the feature vector used is

$$\mathbf{o}_t = [x_t, y_t, p_t, \theta_t, v_t + \Delta + \Delta\Delta]' = [x_t, y_t, p_t, \theta_t, v_t, \dot{x}_t, \dot{y}_t, \dot{p}_t, \dot{\theta}_t, \dot{v}_t, \ddot{x}_t, \ddot{y}_t, \ddot{p}_t, \ddot{\theta}_t, \ddot{v}_t]'. \tag{4.31}$$

The classifiers compared are a BN/GMM model with 30 Gaussian mixture components(929 free parameters), a two-states, 15-Gaussian components HMM (930 free parameters), and a five-states, 6-Gaussian components HMM (933 free parameters). Figure 4.13 shows that results are very slightly worse for the BN/GMM than the HMM classifiers at EER, but not statistically significantly so ($p = 0.05$).

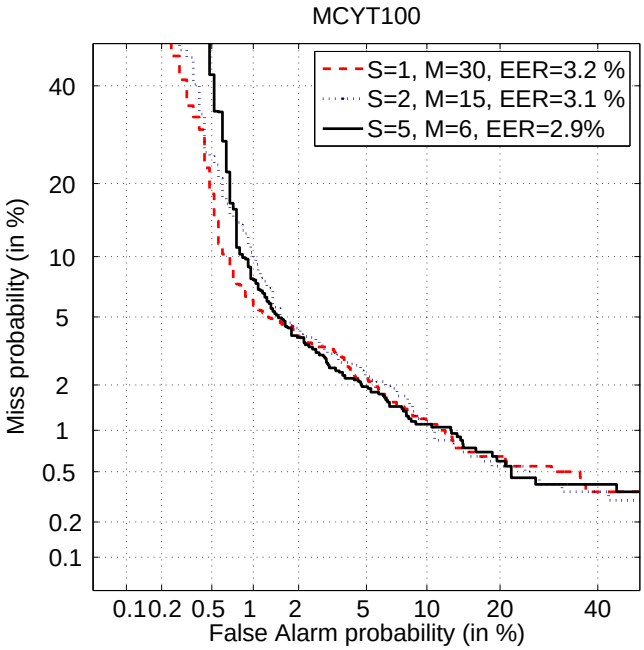


Figure 4.13 — Comparison between the BN/GMM model and HMM models with equivalent number of parameters on MCYT-100.

On SVC 2004, the feature vector used is $\mathbf{o}_t = [x_t, y_t, p_t, \theta_t, v_t, \dot{x}_t, \dot{y}_t, \dot{p}_t, \dot{\theta}_t, \dot{v}_t]'$. The classifiers compared are a BN/GMM model with 50 Gaussian mixture components(1049 free parameters), a two-states, 25-Gaussian components HMM (1050 free parameters), and a five-states, 10-Gaussian components HMM (1053 free parameters). Figure 4.14 shows that results are statistically insignif-

icantly different for these classifiers ($p = 0.05$), and Table 4.5.6 summarises the results in terms of EER.

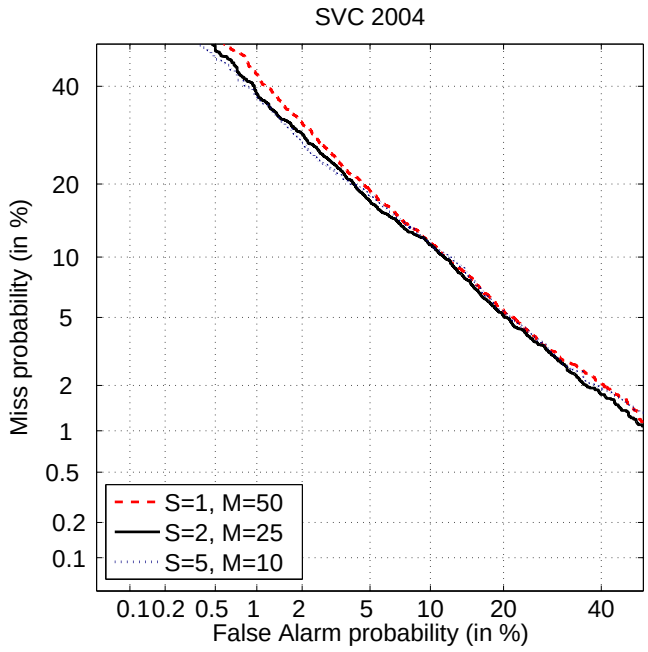


Figure 4.14 — Comparison between the BN/GMM model and HMM models with equivalent number of parameters on SVC2004. Note that the DET curves are computed using the results produced on all 10 folds, hence their smooth aspect.

S	M	EER_{μ} [%]	EER_{σ} [%]	EER_{min} [%]	EER_{max} [%]
1	50	10.8	0.7	9.7	11.8
2	25	10.8	0.9	9.4	12.9
5	10	10.8	0.8	9.5	12.3

Table 4.3 — EER results of the BN/GMM model and the HMM models on the SVC2004 development set, according to the experimental protocol for task 2. EER figures are given over 10 fold cross-validation.

On BMEC 2007, the feature vector used is $\mathbf{o}_t = [x_t, y_t, \dot{x}_t, \dot{y}_t, \ddot{x}_t, \ddot{y}_t]'$. The classifiers compared are a the BN/GMM model with 20 Gaussian mixture components(259 free parameters), a two-states, 10-Gaussian components HMM (260 free parameters), and a five-states, 4-Gaussian components HMM (263 free parameters). Figure 4.15 shows that the difference in error rates between these models is not statistically significant ($p = 0.05$). As a comparison point, the BioSecure reference system (based on HMMs) on the same data achieves 15% EER.

For the databases we have examined, as was posited, the BN/GMM model proposed here performs equivalently to state-of-the art HMM model topologies of equivalent complexity. Upon examination of the most likely path through the model states given by Viterbi decoding, it was found that for most signatures the training data was split regularly according to the number of states. Thus, the 2-states model splits the data into two clusters, corresponding approximately to the first half of the signature and the second half of the signature. In this case, the time-sensitive nature of HMMs may capture some time-dependent specificities, for instance the writing speed is often high

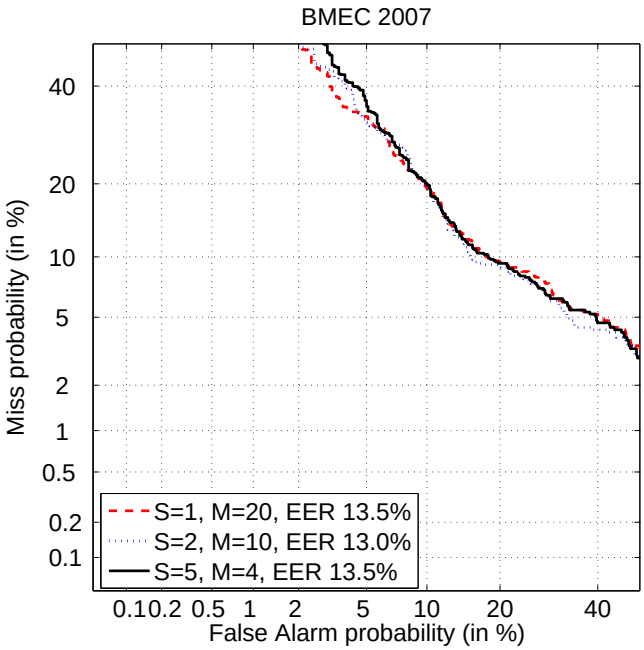


Figure 4.15 — Comparison between the BN/GMM model and HMM models with equivalent number of parameters on BMEC2007.

at the beginning and/or end of a signature. As shown in experimental results, this does not result in significant improvement in performance.

Thus, the difference in performance between signature verification classifiers based on GMMs and those based on HMMs seems to come mostly from other elements in the pattern recognition chain (see Section 1.2), such as the preprocessing, feature extraction and selection, and score normalisation techniques.

4.6 Summary

In this Chapter we have reviewed two possible approaches for modelling multi-dimensional data, namely the vector approach and the scalar approach.

We have shown how to represent Gaussian mixture models as Bayesian networks, and proposed two possible approach for biometric verification applications, leading to two possibilities exist to compute verification scores

We have proposed a Bayesian network topology, equivalent to a GMM, for modelling signatures, and exposed the details of the pattern recognition chain for our proposed approach. We have shown that the same topology can be used for speaker verification, with the difference of background model adaptation, which we do not perform in our signature verification model.

Furthermore, the same model can be used for modelling both local and global signature features, by reducing the number of Gaussians in the model. In the past, global features have generally not been modelled using the same model families as for local features. While global features generally offer inferior performance, their use can be key to increasing diversity in an ensemble of signature verification classifiers. Likewise, the different preprocessing techniques discussed have a knock-on effect on the rest of the feature extraction chain, and are therefore another effective way to increase

diversity without resorting to random subspaces or random sampling methods.

The proposed Bayesian network performs equivalently to a state-of the art approach based on Hidden markov models, which can be implemented as dynamic Bayesian networks. This highlight the point that the temporal aspect of signatures may be less important than the distribution of feature vectors.

Quality measures in biometric verification

5

5.1 Introduction

Many factors conspire to cause verification errors in biometrics. Variability in acquisition conditions, as well as variability of the users' presentations entail a certain level of uncertainty in a classifier's decision. In order to address these issues and improve classification performance, it is crucial to be able to measure phenomena which may be indicative of variability.

A *quality measure* is a measurable indicator of a factor impacting the classifier behaviour, which exhibits a dependency relationship with the classifier output scores and/or classifier decisions. It is jointly modelled with the classifier's scores or decisions in order to improve the verification result or provide estimates of the reliability of the verification result.

In pattern recognition terms, quality measures constitute features. They are used in single-classifier systems (Chapter 6), where they are crucial because they provide additional information which can help a stacked classifier to improve upon the results of both the base classifier and a stacked classifier using only scores or decisions. They are also used in multiple-classifier systems (Chapter 8), where they help explain the relationships between classifiers, leading to probabilistic fusion models that outperform fusion models using only the hard or soft output of classifiers.

In Section 5.2, we propose a systematic division of classifier errors as a motivation to the development of quality measures. In Section 5.3, we propose a way to describe existing quality measures. Section 5.4 is concerned with the evaluation of quality measures. Section 5.5 presents modality-specific quality measures based on different signal processing approaches, and Section 5.6 discusses modality-independent quality measures. Finally, Section 5.7 presents evaluation results for the quality measures presented.

5.2 A short taxonomy of classifier errors

We distinguish three types of classification errors in biometric identity verification: *systematic*, *presentation-dependent*, and *user-dependent*.

Systematic errors are those caused by design problems inherent to the pattern recognition system engineering task. These include wrong assumptions about the form or family of the model used to represent the distributions of features under consideration, poor choice of features leading to excessive overlap between classes, insufficient amount of training data, poor estimation of model parameters (for example insufficient number of iterations, or aggressive variance flooring*), or inadequate decision threshold setting.

Presentation-dependent errors are those caused by unforeseen variability in the signal source. These can be caused by degraded environmental conditions (e.g. ligthing variation for face, specular reflection for iris, additive noise or channel noise for speech, residual fingerprints traces), or by extra variability in a signal (e.g. elastic skin distortion for fingerprints, expression of the face, badly executed signature)

User-dependent errors happen only with certain users that do not fit the otherwise correct assumptions about the user population. This is a well-known problem in biometrics, and one of its incarnations in speaker recognition tasks is called the "Doddington Zoo effect" [71].

Thus, we are interested in developing automated measures of quality that are indicative of potential errors, either systematic, presentation-dependent, or user-dependent.

5.3 A short taxonomy of quality measures

Quality measures can be *modality-dependent* and *modality-independent*. *Modality-dependent* measures (such as "frontalness" in face recognition) are not applicable to other modalities, as they exploit specific domain knowledge that can not be transferred to other signals. *Modality-independent* quality measures (such as distance from score to decision threshold) are more generic and can be exploited across different modalities, but may be dependent on the particular classifier type used.

Quality measures can be *absolute* or *relative*. *Relative* quality measures need reference biometric data, and output a comparison to this reference data taken as a "gold standard" of quality. For instance, correlation with average face is a relative measure of quality. *Absolute* measures do not need reference data, except for initial development of the algorithm. A hybrid approach can also be used, whereby an absolute quality measure is extracted and further normalized by some function of the quality of enrollment data (for instance the geometric mean in [84]).

Lastly, quality measures can be extracted *automatically*, or *hand-labelled* by humans (as in [84]). Hand-labelled quality measures are generally *discrete* with few states, encoding expert opinion on a biometric data sample. Automatically extracted quality measures are generally continuous, but some notable exceptions like the NFIQ quality measure for fingerprints [300] exist. Adler and Dembinsky [3] have reported that hand-labelling and automatic measures do not agree for iris and face modalities.

*a commonly used technique in speech modelling, whereby singularity in covariance matrices is avoided by adding some minimal variance floor to mixture components that have no responsibility due to data sparsity.

5.4 Evaluating quality measures

5.4.1 Visual inspection

Since one aim of using quality measures is to predict verification errors, an important way of looking at quality measures is to plot their distributions with respect to two classes: the class of correct classification decisions, and the class of incorrect classifications, which we denote Decision Reliable: $DR = 1$, respectively $DR = 0$ in the terminology of Chapter 6. These densities can be obtained in several ways, but we recommend kernel-based density estimation, histograms or mixture models because many times these distributions will be asymmetrical and multimodal.

5.4.2 Assuming homoscedasticity of scores

A simplifying assumption that can be made is that the variance of the score is equivalent throughout its range. While this does not hold in practice, it allows for the definition of simple measures of performance for quality measures.

Assuming linearity of relationships with quality measures

Quality measures can be evaluated by measuring their statistical dependence on the scores. Under the assumption of linearity this dependence can be estimated by computing the correlation coefficient between the quality measures QM and scores Sc. Additionally, the linear correlation coefficient between the DR variable and the value of the quality measure gives an indication of the ability of the quality measure to predict errors.

It is also possible to use the mean squared Mahalanobis distance between the distributions of the quality measure for the correct classifier decision and erroneous classifier decision cases, a quantity we denote D_M. Higher Mahalanobis distance between the distributions for correct and erroneous decisions distributions indicates the quality measure is a good predictor of classifier errors, but sports an implicit Gaussian assumption about the distributions.

Not assuming linearity of relationships with quality measures

In real-world data, it seems the relationship between quality measures and scores or classifier errors is not linear. Therefore, we resort to a more sophisticated measure of dependence.

To measure the amount of dependence between two random variables, we use a normalised variant of the mutual information. Mutual information measures the average amount of information that X conveys about Y [185]. It is defined as follows:

$$I(X;Y) \stackrel{\triangle}{=} H(X) - H(X|Y), \tag{5.1}$$

where $H(X)$ is the entropy of random variable X and $H(X|Y)$ is the conditional entropy of X if Y is observed. These are defined as

$$H(X) \stackrel{\triangle}{=} \sum_{x} P(x) log \frac{1}{P(x)}. \tag{5.2}$$

and

$$H(X|Y) \stackrel{\triangle}{=} \sum_{x,y} P(x,y) log \frac{1}{P(x|y)}. \tag{5.3}$$

From equations (5.2) and (5.3) we can write the mutual information in terms of joint and marginal probability distributions as follows:

$$I(X;Y) = \sum_{x,y} P(x,y) log \frac{P(x,y)}{P(x)P(y)}. \tag{5.4}$$

The mutual information is bounded by $0 \leq I(X;Y) \leq min(H(X), H(Y))$. The mutual information if 0 if $X \perp\!\!\!\perp Y$, because in this case the joint $P(x,y) = P(x)P(y)$ and the log term is always 0. The upper bound of the mutual information is potentially very large, because entropies do not have an upper limit. For ease of use in computations and easier interpretability of the measure, we wish to have an upper bound of 1. Thus, we make use of a normalised version of the mutual information defined by Strehl and Ghosh [297]:

$$\bar{I}(X;Y) \stackrel{\triangle}{=} \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}. \tag{5.5}$$

We now prove that as required, the lower bound is 0 and the upper bound is 1. If we have perfectly independent random variables,

$$\begin{aligned} \bar{I}(X;Y) &= \frac{H(X) - H(X|Y)}{\sqrt{(H(X)H(Y)}} \\ &= \frac{H(X) - H(X)}{\sqrt{H(X)H(Y)}} \\ &= 0. \end{aligned} \tag{5.6}$$

If we have perfectly dependent random variables (in the limit both variables are equal), we have

$$\begin{aligned} \bar{I}(X;X) &= \frac{H(X) - H(X|X)}{\sqrt{(H(X)H(X)}} \\ &= \frac{H(X) - 0}{\sqrt{H(X)^2}} \\ &= 1. \end{aligned} \tag{5.7}$$

The difference between normalised mutual information and Pearson correlation coefficient is shown with a few examples in Fig. 5.1.

5.4.3 Not assuming homoscedasticity of scores

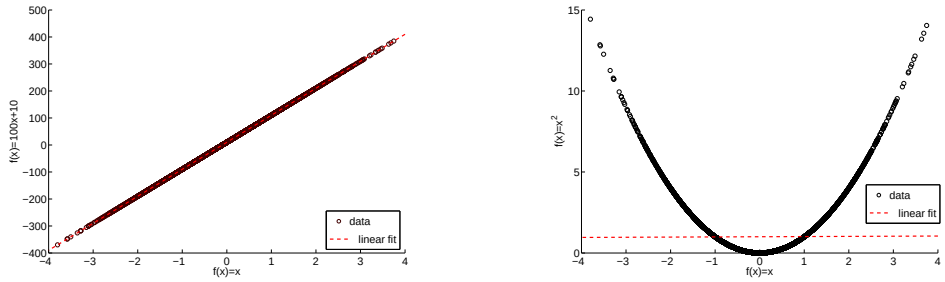
In practice, it is often found that the variance of scores is largely dependent upon the class. We amend our basic performance measures to account for this fact.

Assuming linearity of relationships with quality measures

The partial correlation coefficient [286] is a modification of the classical correlation coefficient in order to compute the correlation between two random variables given knowledge of the state of another random variable.

The (first-order) partial correlation coefficient is defined as:

$$\rho_{xy \cdot z} = \frac{\rho_{xy} - \rho_{xz}\rho_{yz}}{\sqrt{(1 - \rho_{xz}^2)(1 - \rho_{yz}^2)}}, \tag{5.8}$$



(a) Linear relationship. $\bar{I} = 1.0$, $\rho = 1.0$ (b) Non-linear relationship. $\bar{I} = 0.56$, $\rho = 0.01$

Figure 5.1 — Comparison of normalised mutual information $\bar{I}$ and Pearson correlation coefficient ρ for two example linear and non-linear relationships between random variables. The dashed line shows the linear least-squares fit to the data, to provide an graphical view of the Pearson correlation coefficient computation. The data is randomly drawn from a Gaussian distribution.

where the notation $\cdot z$ can be interpreted as "for a subsample where random variable Z has value z". The Z variable is called the control or conditioning variable

To evaluate quality measures, we define two partial correlation coefficients:

$$\rho_{Sc|\Omega} = \rho_{Sc\,QM\,\cdot\Omega} \tag{5.9}$$

$$\rho_{DR|\Omega} = \rho_{DR\,QM\,\cdot\Omega}, \tag{5.10}$$

where $\Omega = \{\omega_0, \omega_1\}$ is the class variable representing either clients ω_1 or impostors ω_0.

Not assuming linearity of relationships with quality measures

If the linearity assumption is not deemed to hold, as is often the case in real-world data, the partial correlation coefficients should be replaced by a (normalised) conditional mutual information measure obtained on the joint densities of interest, either (Sc, QM) or (DR, QM), defined as:

$$I_{Sc|\Omega} = I(Sc; QM|\Omega) \tag{5.11}$$

$$I_{DR|\Omega} = I(DR; QM|\Omega), \tag{5.12}$$

where $\Omega = \{\omega_0, \omega_1\}$ is the class variable representing either clients ω_1 or impostors ω_0.

In this case it is important that the family of densities chosen to model the joint space be either a flexible parametric model (such as a Gaussian mixture model) or a non-parametric variant (such as Parzen windows).

The conditional mutual information allows us to determine what the dependence relationship between any two random variables would be if they were not each dependent on the conditioning random variable. The mutual information between X and Y, with the effects of the conditioning variable Z removed (or equivalently held constant) is given by:

$$I(X; Y|Z) \triangleq H(X|Z) - H(X|Y, Z). \tag{5.13}$$

The conditional mutual information can be written in terms of conditional distributions or marginal distributions as follows:

$$I(X;Y|Z) = \sum_{x,y,z} P(x,y,z) log \frac{P(x,y|z)}{P(x|z)P(y|z)} \quad (5.14)$$

$$= \sum_{x,y,z} P(x,y,z) log \frac{P(z)P(x,y,z)}{P(x,z)P(y,z)}. \quad (5.15)$$

The conditional mutual information is bounded by $0 \leq I(X;Y|Z) \leq H(X)$. Again, to have a an upper bound of 1 we propose a normalised conditional mutual information:

$$\bar{I}(X;Y|Z) \stackrel{\triangle}{=} \frac{I(X;Y|Z)}{\sqrt{H(X|Z)H(Y|Z)}}. \quad (5.16)$$

We now prove that as required, the lower bound of the normalised conditional mutual information is 0 and the upper bound is 1. If we have independent random variables $X \perp\!\!\!\perp Y$,

$$\bar{I}(X;Y|Z) = \frac{H(X|Z) - H(X|Y,Z)}{\sqrt{H(X|Z)H(Y|Z)}}$$

$$= \frac{H(X|Z) - H(X|Z)}{\sqrt{H(X|Z)H(Y|Z)}}$$

$$= 0. \quad (5.17)$$

If we have dependent random variables $X \not\perp\!\!\!\perp Y$, where in the limit both variables are equal:

$$\bar{I}(X;X|Z) = \frac{H(X|Z) - H(X|X,Z)}{\sqrt{(H(X|Z)H(X|Z)}}$$

$$= \frac{H(X|Z) - 0}{\sqrt{H(X|Z)^2}}$$

$$= 1. \quad (5.18)$$

To measure the amount of separation between erroneous decisions $DR = 0$ and correct decisions $DR = 1$ provided by the quality measure, we can use the symmetric Kullback-Leibler divergence between the DR-conditional likelihoods $P_0(QM) = P(QM|DR = 0)$ and $P_1(QM) = P(QM|DR = 1)$:

$$D_{KL}(P_0||P_1) = \sum_{qm} P_0(qm) log\Big(\frac{P_0(qm)}{P_1(qm)}\Big) + \sum_{qm} P_1(qm) log\Big(\frac{P_1(qm)}{P_0(qm)}\Big). \quad (5.19)$$

We use Gaussian mixture models with diagonal covariances and 3 Gaussian components for estimating the densities used for D_{KL}. Because of the stochastic initialisation for the parameters of each component density, we run the divergence computation several times and take an average value.

5.4.4 The impact of background modelling

As mentioned in Chapter 4, background models are often used in biometric verification for modalities such as speech, signature, or face. For modalities that are susceptible to degradation of acquisition conditions, it is often claimed that using a background model somewhat compensates for mismatch in conditions, since the background model itself will have been trained in nominal conditions, thus suffering similar distortions in *likelihood* than the user model [12].

While we do not dispute the effectiveness of background modelling to reduce the effect of noise, we contend that the effects of background model normalisation may be different for clients and impostors *scores*. An example is shown in Figure 5.2, where a quality measure related to the signal-to-noise ratio is plotted against score distributions. It clearly appears that client scores are more correlated with the quality measures than impostor scores.

In addition to our own experiments, evidence to support this claim is found in numerous publications on speaker recognition. For instance, in [162, Figure 5] it is apparent that the client score distribution is much more affected by mismatched transmission channels than the impostor score distribution. In [100, Figure 1], the addition of artificial Gaussian noise on the speech modality affects the client distribution much more than the impostor distribution. In [119, Figures 2-3], the client distribution is again more perturbed than the impostor distribution when tested with different handsets.

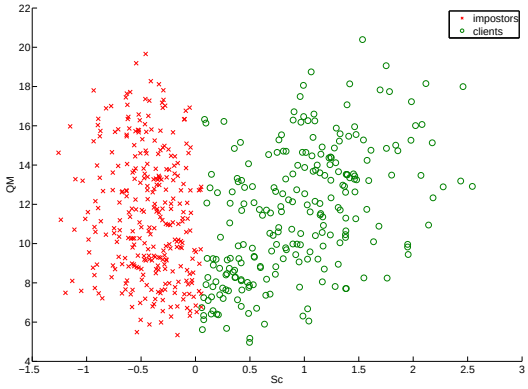


Figure 5.2 — Scatterplot of scores and a SNR-related quality measure showing different correlations depending on class due to background modelling. Crosses indicate impostors and circles indicate clients

We now proceed to prove that it is possible that noise has a different effect on clients and impostor scores. In order to do so, we assume that the signal is contaminated by unspecified noise having a linear effect (shift) in the likelihood domain. The clean likelihoods

$$\mathcal{L}_{\cdot} = P(\mathbf{O}; \mathbf{\Theta}^{\cdot}) \geq 0 \tag{5.20}$$

are contaminated by noise and become noisy likelihoods $\widetilde{\mathcal{L}}_{\cdot}$:

$$\widetilde{\mathcal{L}}_{\cdot} = \mathcal{L}_{\cdot} + \epsilon, \tag{5.21}$$

where ϵ represents the effect of noise on the likelihood, a quantity that can be either negative or positive. Since we cannot suppose that the noise behaves differently for client attempts and impostor attempts, the noisy likelihoods for clients $\widetilde{\mathcal{L}}_{\omega_1}$ (obtained by comparing genuine biometric data to a user model), impostors $\widetilde{\mathcal{L}}_{\omega_0}$ (obtained by comparing impostor data to a user model), and those obtained on the world model $\widetilde{\mathcal{L}}_{-}$ obey Equation (5.21).

The verification scores are computed by taking the log of the likelihood ratios, thus for the clean case we have:

$$Sc_{\cdot} = log\left(\frac{\mathcal{L}_{\cdot}}{\mathcal{L}_{-}}\right), \tag{5.22}$$

and for the noisy case we have:

$$\widetilde{Sc}_{\cdot} = log\big(\frac{\widetilde{\mathcal{L}}_{\cdot}}{\widetilde{\mathcal{L}}_{-}}\big). \tag{5.23}$$

We define a class- and noise-dependent function $\Delta(\epsilon, \omega_{\cdot})$, which quantifies the amount the score for class $\omega_{\cdot}$ is shifted when subjected to noise ϵ:

$$\Delta(\epsilon, \omega_{\cdot}) \stackrel{\triangle}{=} \widetilde{Sc}_{\omega_{\cdot}} - Sc_{\omega_{\cdot}}. \tag{5.24}$$

Our hypothesis, namely that noise can have a different effect on impostor and client score, is now formulated as

$$\exists\ \epsilon \text{ such that } \quad \Delta(\epsilon, \omega_0) \neq \Delta(\epsilon, \omega_1). \tag{5.25}$$

We can find conditions under which this hypothesis holds by expanding the alternative hypothesis:

$$\begin{aligned}
\Delta(\epsilon, \omega_0) &= \Delta(\epsilon, \omega_1) \\
log(\mathcal{L}_{\omega_0}) - log(\mathcal{L}_{-}) - log(\mathcal{L}_{\omega_0} + \epsilon) + log(\mathcal{L}_{-} + \epsilon) &= log(\mathcal{L}_{\omega_1}) - log(\mathcal{L}_{-}) - log(\mathcal{L}_{\omega_1} + \epsilon) + log(\mathcal{L}_{-} + \epsilon) \\
log(\mathcal{L}_{\omega_0}) - log(\mathcal{L}_{\omega_0} + \epsilon) &= log(\mathcal{L}_{\omega_1}) - log(\mathcal{L}_{\omega_1} + \epsilon) \\
log\Big(\frac{\mathcal{L}_{\omega_0}}{\mathcal{L}_{\omega_0} + \epsilon}\Big) &= log\Big(\frac{\mathcal{L}_{\omega_1}}{\mathcal{L}_{\omega_1} + \epsilon}\Big)
\end{aligned} \tag{5.26}$$

By definition of a functioning pattern classifier, we must have on average higher likelihoods for samples from the class than for samples not from the class:

$$\bar{\mathcal{L}}_{\omega_1} > \bar{\mathcal{L}}_{\omega_0}. \tag{5.27}$$

Further taking into account the lower bound on Eq. (5.20), function analysis shows that the nature of the relationship between client scores and impostor scores under noise depends on ϵ:

$$\begin{aligned}
&\Delta(\epsilon, \omega_0) > \Delta(\epsilon, \omega_1) \quad \text{if } \epsilon < 0 \\
&\Delta(\epsilon, \omega_0) = \Delta(\epsilon, \omega_1) \quad \text{if } \epsilon = 0 \\
&\Delta(\epsilon, \omega_0) < \Delta(\epsilon, \omega_1) \quad \text{if } \epsilon > 0
\end{aligned} \tag{5.28}$$

Hence, on average, for non-zero ϵ, noise that manifests itself as a linear shift in the likelihood domain can affect clients and impostor score distributions differently.

Therefore, the evaluation of modality-specific quality measures that are used together with pattern recognition systems using a log-likelihood ratio scoring technique (such as those described in Chapter 4) must take into account the class (impostor or client) of the access with which the quality measure is associated. This is a further argument in favour of using class-conditional evaluation methods such as partial correlation coefficient or conditional mutual information.

5.4.5 A feature selection perspective

An important point is that the ultimate evaluation for a quality measure is to apply it to a biometric verification task dataset and see if it leads to improvements in terms of final error rate or rejection rate. While a quality measure may seem to poorly separate the error-conditional distributions, for instance as pointed out by a low Mahalanobis distance, there may still exist a classifier which can make use of the quality data.

This is analogous to the situation in feature selection, where filter methods (functions indicative of the ultimate performance) are generally found to provide inferior results to wrapper methods [141], where the measure of performance is the use of a feature with the classifier itself.

5.5 Modality-specific measures

Modality-specific measures can account for degradation in signal quality. In speech processing, we can use both time-domain techniques and spectral-domain techniques to obtain a quantity correlated with the amount of noise in the signal.

5.5.1 Quality measures based on speech segmentation in the time domain

Voice activity detection (VAD), also called speech/pause segmentation, can be used to obtain an estimate of the signal-to-noise ratio. This is done by assuming the average energy in pauses represents the noise energy, and the energy in speech represents the signal energy. The formulation for this family of quality measures is:

$$QM_{VAD.} = 10\log_{10}\Big(\frac{\sum_{t=1}^{T} Is(t)s^2(t)}{\sum_{t=1}^{T} In(t)s^2(t)}\Big), \tag{5.29}$$

where $\{s(t)\}, t = 1, \ldots, T$ is the acquired speech signal containing T samples, $Is(t)$ and $In(t)$ are the indicator functions of the current sample $s(i)$ being speech or noise during pauses (e.g. $Is(t)$=1 if $s(t)$ is a speech sample, $Is(t)$=0 otherwise) as reported by the voice activity detector.

We perform voice activity detection by using two different algorithms, one based on the energy of the signal, the other based on the spectral entropy. The SNR estimated then yields two quality measures, respectively QM_{VAD_E} and QM_{VAD_H}.

Energy-based VAD

The "Murphy algorithm" [245] is based on approximating the noise by a time-varying lowpass-filtered signal energy. The noise energy is initialised to the average energy of the samples in the first frame, then adapted to follow variations of the noise by implementing three heuristics (slowly increase the estimated noise energy unless it's above the lowpass-filtered signal energy or above twice the energy in the current frame). The signal-to-noise ratio is estimated at each frame, and frames are labelled as speech when an SNR threshold is exceeded.

The main problem with this VAD is that it is sensitive to noise: indeed, the main assumption used to differentiate speech from noise is that the energy is higher in speech portions of the signal. By definition, this assumption does not hold in negative SNRs, and the algorithm performance degrades rapidly with increasing amounts of noise. Thus, we use a second segmentation algorithm based on spectral entropy.

Entropy-based VAD

It is also possible to use the short-term spectral entropy for performing voice activity detection and assigning values to $Is(t)$ and $In(t)$*. The entropy is a measure defined over a probability distribution function, which measures the informativeness of the distribution. The spectral entropy is calculated over the short term spectrum values, where the spectral values are normalized to sum up to 1, thus forming a pdf.

*Yet another possibility is to compute the signal-to-noise ratio directly in the spectral domain [78].

The spectral entropy at frame t is computed as follows:

$$H(|Y(t)|^2) = -\sum_{w=1}^{\Omega} \frac{|Y(w,t)|^2}{\Sigma_{w=1}^{\Omega}|Y(w,t)|^2} \log\left(\frac{|Y(w,t)|^2}{\Sigma_{w=1}^{\Omega}|Y(w,t)|^2}\right), \tag{5.30}$$

where $|Y(w,t)|^2$ is the power spectrum in frequency band ω for frame t.

$H(|Y(t)|^2)$ is maximised when we have white noise and is minimised when we have a pure tone. The application of entropy relies on the assumption that the presence of pitch in speech segments results in a more organised signal (presenting series of peaks in the spectrum) compared with the case of noise (pauses). Thus, the entropy value is higher for pause than speech regions. We then fit a Gaussian mixture model with two components (one for speech regions, one for pause regions) on the distribution of spectral entropies using the EM algorithm (see Section 3.3.1), and choose the entropy threshold according to the Bayesian decision rule (Equation (1.2)).

The algorithm we use is based on the work by Renevey and Drygajlo [244].

5.5.2 Quality measures based on higher-order statistics

Since clean speech has a very distinctive distribution (sharp peak at sample value 0 - a large amount of a speech signal is actually silence), we can exploit this knowledge to infer when the signal is noisy. The additive noise we are concerned about has energy (if it does not then it does not impair the speech signal), which means it will contribute to modifying the time-domain distribution of amplitudes. This is illustrated in Fig. 5.3.

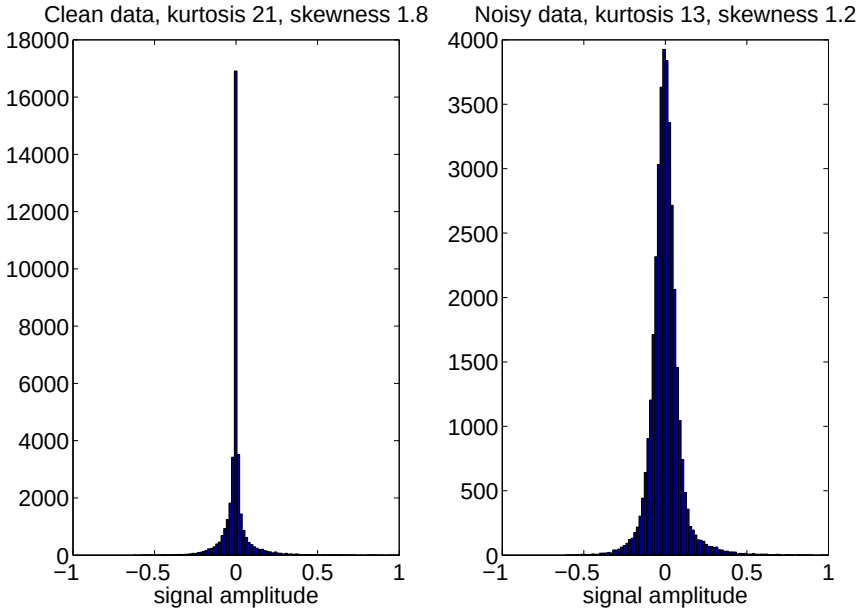


Figure 5.3 — Histogram of time-domain signal amplitudes for a clean and noisy (babble-type additive noise) TIMIT utterance.

Higher order statistics can be used to summarise the shape of unimodal distributions in a meaningful way. The skewness (or Fisher skewness) measures the asymmetry of a distribution with respect to its mode. Any symmetrical distribution (such as Laplace, Gaussian, or uniform) has a skewness of 0. Negative skewness indicates that the distribution has a longer tail on the left of the

mode, while positive skewness indicates the opposite. Skewness is defined as

$$QM_{skew} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{s_t - \mu_s}{\sigma_s}\right)^3, \tag{5.31}$$

where s_t is a signal sample at time t, μ_s is the signal mean, and σ_s is the signal standard deviation.

Kurtosis (or Fisher kurtosis), defined in Eq. (5.32), corresponds to the "peakiness" of the distribution. By definition, a Gaussian distribution has a kurtosis of 3*. A leptokurtic (or supergaussian) distribution has a kurtosis higher than 3 and is "peakier", while a platykurtic (or subgaussian) distribution has a kurtosis lower than 3 and is "flatter", that is its probability density is spread over a larger dynamic input range. Therefore, it is probable that a noisy speech distribution has a lower kurtosis than a clean speech distribution. This was exploited by Nemer et al. [207] for estimating SNR based on subband decomposition.

$$QM_{kurt} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{s_t - \mu_s}{\sigma_s}\right)^4 \tag{5.32}$$

Unfortunately, kurtosis estimation is very sensitive to outliers. We therefore introduce a third related measure, called the centre bin measure, to approximate kurtosis and estimate the peakiness of the distribution. First, the signal sample amplitudes are binned in 100 equally-spaced bins, then the measure is defined as the ratio of the number of samples in the bin containing the most samples to the total number of samples in the other bins.

$$QM_{bin} = \frac{N_{max}(s)}{\left(\sum_B N_b(s)\right) - N_{max}(s)}, \tag{5.33}$$

where $N_b(s)$ represents the number of samples in bin b, and $N_{max}(s)$ represents the number of samples in the bin that contains the most samples.

5.5.3 (lack of) Signal-domain quality measures for signature

In signature verification, the data is digitised directly from the pen tablet. Environmental changes do not affect the signal, and the only noise present is the unavoidable quantisation noise. Except in case of sensor failure or acquisition software problems †, signature data can be said to be noise-free.

While most research in signal-domain quality measures for signature has concentrated on measures of intra-user variability (see Section 2.5), in our case the variability is handled by the probabilistic base classifier. Thus, we use only modality-independent quality measures for signature verification.

5.6 Modality-independent quality measures

Keeping in mind that the goal of quality measures is to help predict verification errors, we can use some information that does not directly depend on the underlying signal properties. Here we review three approaches that are generic enough to be used with many modalities and classifiers, though each approach may need to be adapted to fit different classifier families.

*Or 0, as some definitions of kurtosis subtract 3 to have kurtosis of 0 for the normal distribution.

†such as caused by scheduling or buffering problems in low-power mobile devices

5.6.1 Score-based

Many classifiers provide a continuous-valued output (measurement-level) indicating how close a particular sample is to a particular class, a quantity called *score* in biometrics. The probability of classification error increases as the score gets closer to the decision boundary between classes. This "soft" classifier output, and its distribution constitute valuable data for error prediction, and are applicable to any biometric modality whose classifier produces a non-discrete output. The score is the base of many confidence models (See Section 2.4 and e.g. [20, 107, 201, 232]).

Quantities derived from the score are also used, for instance variance of the score (provided by human expert knowledge of the problem domain) and distance from normalized score to "hard" (decision-level) classifier output* [22]. Indeed, the distance from the score to the decision threshold constitutes a quality measure: it is more probable that the classifier will make a mistake if a score is close to the decision boundary, as noise alone could have moved that score over the threshold. This is the idea behind the method of margins [232].

The distance from user-specific to user-independent decision threshold can be used as a quality measure. In a verification system with a user-independent threshold†, some users will be more systematically subjected to false rejects, respectively false accepts, than others. Combining this quality measure with the score improves the results on the subsequent classification or regression task [257].

5.6.2 User model-based quality measures

Information about the user models can be used to detect systematic errors. For instance, the quality of parameter estimation can be taken into account. In modalities such as signature, where no environmental noise is present, this consitutes precious additional information for classification.

Formally, if we assume an infinite amount of non-distorted training data and knowledge of the correct form of the parametric density function, Maximum Likelihood training will result in asymptotically correct model parameters $\mathbf{\Theta}^{u\circ}$ for each user. In this case, further assuming i.i.d. testing data, the application of the Bayesian decision rule (Equation (1.2)) based on the likelihood computation $P(\mathbf{O};\mathbf{\Theta}^{u\circ})$ will give optimal classification results. However, in practical learning we cannot compute $\mathbf{\Theta}^{u\circ}$ and must be satisfied with $\mathbf{\Theta}^{u}$, a distorted version of $\mathbf{\Theta}^{u\circ}$ and the corresponding distorted likelihood $P(\mathbf{O};\mathbf{\Theta}^{u})$‡. User model-based quality measures provide observable data related to this parametric distortion.

In the case of statistical models such as GMMs, the distance (likelihood) computation rests upon the Mahalanobis distance between the user's model (mean vectors $\boldsymbol{\mu}^u$, covariance matrices $\mathbf{\Sigma}^u$, and mixing coefficients) and the biometric pattern. The Mahalanobis distance for a Gaussian component is expressed as follows (component index m is dropped to favour legibility):

$$d_{Mahal} = (\mathbf{o} - \boldsymbol{\mu}^u)' \mathbf{\Sigma}^{u^{-1}} (\mathbf{o} - \boldsymbol{\mu}^u). \tag{5.34}$$

As can be seen from Eq. (5.34), this distance requires an inversion of the covariance matrix $\mathbf{\Sigma}^u$. Because this covariance matrix is typically estimated from a limited amount of data using a maximum likelihood procedure, it may be ill-conditioned, meaning that the quality of inversion will be low, which in turn entails errors in the Mahalanobis distance computation.

*assuming the classifier decisions are the integer extremal points in the score interval, which is typically $[0, 1]$

†For instance because it has recently been deployed and there is not enough data for each user to reliably set a personalised threshold.

‡In the terminology of Kharin [147], this corresponds to an "error in parameter assignment"

We propose two quantities to estimate the "stability" of the covariance matrix of a single Gaussian component under numerical operations such as inversion: the first is the *determinant*, and the second is the *condition number*.

By definition, a matrix is invertible only if its determinant is non-zero [8]. If the determinant for a covariance matrix is close to zero, the matrix may be badly conditioned. Whether the numerical algorithm uses Gauss-Jordan elimination or another approach to inverse the matrix, the inversion may be erroneous. Thus, the closer to zero the determinant of a covariance matrix is, the more biased the result of Mahalanobis distance will be, and the more likely it is that the classification will be wrong. For Gaussian mixture component m and user model u, the quality measure is defined as

$$QM_{det_m^u} = |\mathbf{\Sigma}_m^u|. \tag{5.35}$$

Another quality measure we propose to use is the *condition number*, which is defined as the ratio of the norm of the covariance matrix to the norm of the inverse covariance matrix. We use the 2-norm, which on a square matrix (as is the case for covariance matrices) is equivalent to the largest singular value of the matrix. The condition number quality measure for Gaussian component m is

$$QM_{\mathcal{K}_m^u} = \|\mathbf{\Sigma}_m^u\|_2 \cdot \|\mathbf{\Sigma}_m^{u^{-1}}\|_2. \tag{5.36}$$

A large $QM_{\mathcal{K}_m^u}$ indicates an ill-conditioned matrix, wherease a value of 1 indicates a well-conditioned matrix.

Since we generally use mixtures of Gaussian densities for modelling the biometric data of users, it is necessary to aggregate the quality measures of individual densities into a single quality measure for each model. We use three aggregation methods: summing (denoted $QM_{\cdot}$), averaging (denoted $\overline{QM}_{\cdot}$), and weighted sum (denoted $\overline{QM}_{\cdot_w}$). In weighted aggregation, the weights come from the mixing coefficients c_m assigned to each Gaussian mixture component. This is done because Gaussian components with low mixing coefficients will not have a large impact on the likelihood output, and thus their potential low quality is less damaging to the overall Mahalanobis distance computation. Furthermore, we also propose to only take into account the M_{max} components with the "worst" quality measures. That is, only the components with the lowest determinant quality measures, and those with the highest condition number. The summed versions of the quality measures are computed as

$$QM_{\cdot} = \sum_{m=1}^{M_{max}} QM_{\cdot_m}, \tag{5.37}$$

where the $QM_{\cdot_m}$ are sorted in descending order for $QM_{\mathcal{K}_m}$ and in ascending order for QM_{det_m}. M_{max} can be set to M in order to compute the aggregated quality measure over all Gaussian densities, or to some arbitrary proportion of M. The averaged versions of the quality measures are computed as

$$\overline{QM}_{\cdot} = \frac{1}{M_{max}} \sum_{m=1}^{M_{max}} QM_{\cdot_m}. \tag{5.38}$$

Finally, the weighted sum versions of the quality measures are computed as

$$\overline{QM}_{\cdot_w} = \sum_{m=1}^{M_{max}} \frac{1}{c_m} QM_{\cdot_m}. \tag{5.39}$$

Since the $QM_{det_m^u}$ can be numerically small and the $QM_{\mathcal{K}_m^u}$ can be numerically large, in practical implementations we take the logarithm of the aggregated quality measures.

The problem of adaptation

Depending on the adaptation scheme used in model training, quality measures based on the covariance matrix may result in the same values for all users, thus not providing useful information for single-classifier systems.

If the user models $\mathbf{\Theta}^u$ are MAP-adapted from the world model $\mathbf{\Theta}^w$ by adapting *only the means*, then the Mahalanobis distance computation for one single Gaussian component is:

$$d_{Mahal} = (\mathbf{o} - \boldsymbol{\mu}^u)' \mathbf{\Sigma}^{w^{-1}} (\mathbf{o} - \boldsymbol{\mu}^u), \tag{5.40}$$

where the covariance matrix used in the computation is the covariance matrix of the world model $\mathbf{\Sigma}^w$, since in this adaptation scheme we have:

$$\forall\, u, \mathbf{\Sigma}^u = \mathbf{\Sigma}^w. \tag{5.41}$$

Other model scoring approaches

Penalty-based scoring functions for model selection, such as MDL (see Equation (4.26)) or AIC can also be used to obtain a user-model quality measure. Interestingly, for a BN/GMM signature verification classifier using global features, the final log-likelihood obtained after convergence of the EM algorithm is an offset version of MDL, and can directly be used as a quality measure, although with the caveats mentioned in Section 4.5.4.

5.7 Experiments and results

For these experiments, we evaluate the performance of quality measures with respect to different databases by using the indicators defined in Section 5.4. We repeat them here for reference.

$\rho_{Sc|\omega_0}$ and $\rho_{Sc|\omega_1}$ are the class-conditional linear correlation coefficients between the classifier score output and the quality measure for impostors and clients respectively. $\rho_{DR|\omega_0}$ and $\rho_{DR|\omega_1}$ are the class-conditional linear correlation coefficients between the quality measure and the decision correctness indicator (DR) for impostors and clients respectively.

$I_{Sc|\omega_0}$ and $I_{Sc|\omega_1}$ are the normalised conditional mutual informations between the classifier score output and the quality measure for impostors and clients, and $I_{DR|\omega_0}$ and $I_{DR|\omega_1}$ are the normalised conditional mutual informations between the quality measure and the decision correctness indicator (DR) for impostors and clients respectively. D_{KL} is the symmetric Kullback-Leibler divergence between the DR-conditional distributions of quality measures.

5.7.1 Modality-independent quality measures

In these experiments, we evaluate the performance of the quality measures defined in Section 5.6 over the MCYT-100, SVC2004, and BMEC 2007 signature databases. The classifiers are the BN/GMM models described in Section 4.5.6.

The results are shown in Table 5.1, Table 5.2, and Table 5.3

Overall, it appears that the quality measures based on the determinant of the covariance matrix have both higher correlation and higher mutual information with classifier scores than those based on the condition number. In this case, modelling the relationship between the quality measures and the scores is likely to bring benefits to tasks such as fusion. In terms of error prediction, the very low correlations, mutual informations, and Kullback-Leibler divergences point to the fact that these quality measures will not be very useful in predicting error, at least with simple models.

measure	$\rho_{Sc\|\omega_0}$	$\rho_{Sc\|\omega_1}$	$I_{Sc\|\omega_0}$	$I_{Sc\|\omega_1}$	$\rho_{DR\|\omega_0}$	$\rho_{DR\|\omega_1}$	$I_{DR\|\omega_0}$	$I_{DR\|\omega_1}$	D_{KL}
$QM_{\mathcal{K}}$	0.18	-0.16	0.06	0.06	0.06	0.08	0.01	0.02	0.21
$\overline{QM}_{\mathcal{K}_w}$	0.14	-0.16	0.04	0.09	0.04	0.04	0.02	0.01	0.13
QM_{det}	0.50	-0.37	0.12	0.14	-0.02	0.06	0.02	0.02	0.23
$\overline{QM}_{det_w}$	0.50	-0.37	0.11	0.13	-0.02	0.06	0.02	0.02	0.24

Table 5.1 — Performance of modality-independent quality measures on the MCYT-100 dataset.

measure	$\rho_{Sc\|\omega_0}$	$\rho_{Sc\|\omega_1}$	$I_{Sc\|\omega_0}$	$I_{Sc\|\omega_1}$	$\rho_{DR\|\omega_0}$	$\rho_{DR\|\omega_1}$	$I_{DR\|\omega_0}$	$I_{DR\|\omega_1}$	D_{KL}
$QM_{\mathcal{K}}$	-0.11	-0.20	0.09	0.09	0.08	-0.10	0.03	0.04	0.16
$\overline{QM}_{\mathcal{K}_w}$	-0.11	-0.20	0.09	0.09	0.08	-0.10	0.03	0.04	0.16
QM_{det}	-0.06	-0.16	0.10	0.08	0.05	-0.09	0.04	0.04	0.16
$\overline{QM}_{det_w}$	-0.05	-0.15	0.10	0.08	0.05	-0.08	0.05	0.04	0.10

Table 5.2 — Performance of modality-independent quality measures on the SVC2004 dataset.

5.7.2 Modality-specific quality measures

The first database used is the speech part of XM2VTS, following the Lausanne protocol, configuration 1. Where applicable, the results are reported by training the models on the evaluation set and testing them on the testing set.

The second database is the speech part of the BANCA database, P protocol. Where applicable, the results are reported by taking an average of measures when first training the fusion model on G1 and testing on G2, then training on G2 and testing on G1.

Additionally, to evaluate the performance of speech segmentation, on which the QM_{VAD} family of quality measures is based, we use the CUAVE audio-visual database [223].

The speaker verification system used for BANCA is described in Section 4.4.5.

On XM2VTS, we use the 200-Gaussian components GMM classifier from [233], which uses 16 spectral subband centroid features.

Energy-based quality measure for speech: performance of speech segmentation

Since the SNR estimate depends on the speech/pause segmentation, we evaluated the performance of this VAD on the "individuals" set of the CUAVE database [223]. The performance is computed in terms of four quantities [89]: *front-end clipping* (FEC), indicating speech missclassified as noise due to the transition from noise to speech. *Mid-speech clipping* (MSC) indicates speech misclassified as noise during a speech period. Noise classified as speech when the signal transitions from speech to noise is denoted $OVER$. Finally, noise that is classified as speech during a noise period is denoted NDS. We simplify the evaluation of performance by reporting 3 joint quantities: noise classified as

measure	$\rho_{Sc\|\omega_0}$	$\rho_{Sc\|\omega_1}$	$I_{Sc\|\omega_0}$	$I_{Sc\|\omega_1}$	$\rho_{DR\|\omega_0}$	$\rho_{DR\|\omega_1}$	$I_{DR\|\omega_0}$	$I_{DR\|\omega_1}$	D_{KL}
$QM_{\mathcal{K}}$	0.28	-0.21	0.11	0.08	0.03	-0.15	0.05	0.03	0.14
$\overline{QM}_{\mathcal{K}_w}$	0.22	-0.31	0.10	0.11	0.03	-0.25	0.03	0.10	0.12
QM_{det}	0.32	-0.40	0.12	0.11	0.10	-0.15	0.04	0.02	0.12
$\overline{QM}_{det_w}$	0.33	-0.40	0.11	0.11	0.09	-0.15	0.04	0.03	0.14

Table 5.3 — Performance of modality-independent quality measures on the BMEC 2007 signature dataset.

speech ($NAS = OVER + NDS$), speech classified as noise ($SAN = FEC + MSC$), and total error rate R which is the number of signal samples missclassified, no matter whether they were speech or noise. These three quantities are evaluated for each file in the CUAVE database (36 files) and the average is presented in Table 5.4. It should be noted that the majority of errors are made on three particular files (subjects), and that the files have a high signal-to-noise ratio. Therefore, the VAD will be less accurate on noisy data. This confirms that it could prove useful to combine quality measures derived from this speech/pause segmentation with other quality measures, especially if they are robust to noise (see Section 5.7.2).

NAS_μ [%]	SAN_μ [%]	R_μ [%]
13.03	11.45	12.47

Table 5.4 — Percentage of noise samples classified as speech (NAS_μ), percentage of speech samples classified as noise (SAN_μ), and total classification error (R_μ). All results are averaged over the utterances in the individuals set of the CUAVE database.

Energy-based quality measure for speech: performance of SNR estimation

To evaluate the correlation of the energy-based quality measure QM_{VAD_E} with a known signal-to-noise ratio, we run the algorithm against the noisy version of XM2VTS described in Appendix A.1, thus producing a set (real SNR, quality measure) for each utterance. The results are shown in Fig. 5.4. Here, assuming a linear relationship, it can be seen that the energy-based measure is highly correlated ($\rho = 0.82$) with the real signal-to-noise ratio. Thus, it can be expected to be a good indicator of babble-type additive noise.

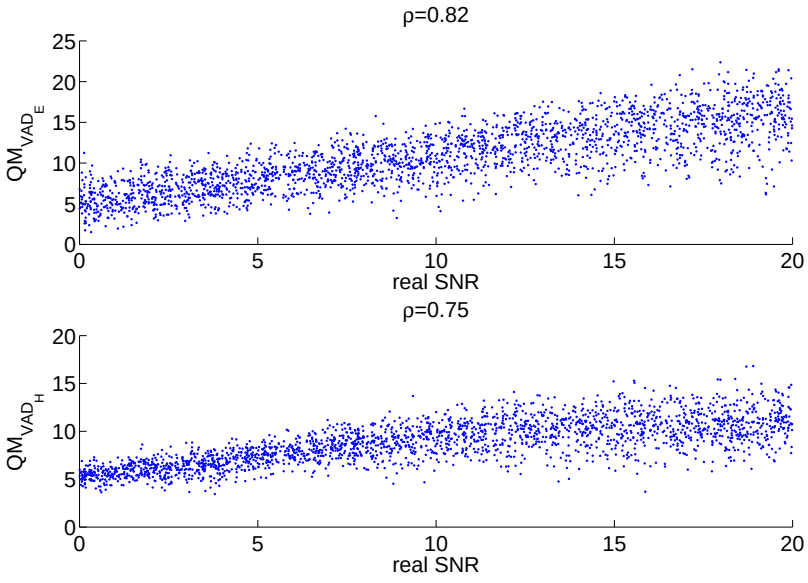


Figure 5.4 — Correlation between the energy-based QM_{VAD_E} signal quality measure and the entropy-based signal quality measure QM_{VAD_H} and real signal-to-noise ratio on a noisy version of the evaluation subset of XM2VTS. Each data point corresponds to an utterance.

The distribution of the energy-based quality measure on BANCA is shown in Fig. 5.5. It is

important to model this quality measure as a mixture distribution, or the bimodal nature of the "correct decisions" distribution will be poorly estimated.

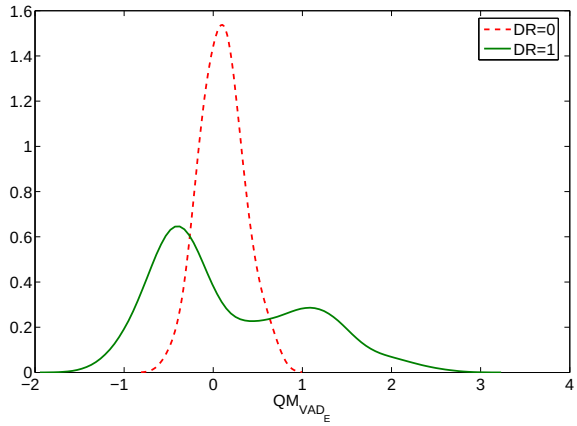


Figure 5.5 — Distributions of energy-based quality measure QM_{VAD_E} for correct (DR=1) and erroneous (DR=0) classifier decisions on BANCA G1 data.

Entropy-based quality measure for speech: performance of SNR estimation

We run the algorithm described in Section 5.5.1 against the noisy version of XM2VTS described in Appendix A.1, thus producing a set (real SNR, quality measure) for each utterance. The results are shown in Fig. 5.4. Assuming a linear relationship, the entropy-based measure is also highly correlated ($\rho = 0.75$) with the real signal-to-noise ratio. The superior performance of this estimator in very noisy conditions (SNR=5 dB or below) with respect to the energy-based quality estimator is made clear from this figure, where it can be seen that the spread of estimates for this SNR range is much lower than that of the energy-based quality estimator*.

The distribution of the entropy-based quality measure on BANCA is shown in Fig. 5.6. Here, in general, and according to intuition, higher values of SNR mean higher signal quality and fewer errors, something to be contrasted with the energy-based measure.

Higher order statistics measures of quality for speech: performance of SNR estimation

To evaluate the correlation of the quality measures with the real signal-to-noise ratio, we again use the noisy XM2VTS database of Appendix A.1. The results for kurtosis (Eq.(5.32)), skewness (Eq.(5.31)), and the centre bin measure (Eq.(5.33)) are shown in Fig. 5.7. Here it can be seen that the centre-bin measure is highly correlated ($\rho = 0.54$) with the real signal-to-noise ratio. Thus, it can be expected to be a good indicator of babble-type additive noise.

However, good correlation with signal-to-noise ratio does not guarantee that we will be able to predict errors, as the models or features may be somewhat robust to the kind of noise measured. Also, it is probable that the best quality measure on a particular database is not the same for other database, where the noise characteristics may be very different. We therefore plot the DR-dependent distributions of quality measures in Fig. 5.8 for BANCA G1 data.

*Numerically, the residuals for a least-square linear fit are much smaller. While this heteroscedasticity means that a linear correlation coefficient should not be used, we provide the correlation figure as a rough approximation.

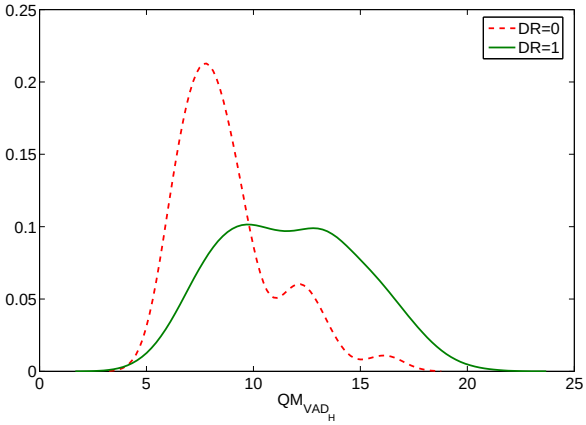


Figure 5.6 — Distributions of entropy-based quality measure QM_{VAD_H} for correct (DR=1) and erroneous (DR=0) classifier decisions on BANCA G1 data.

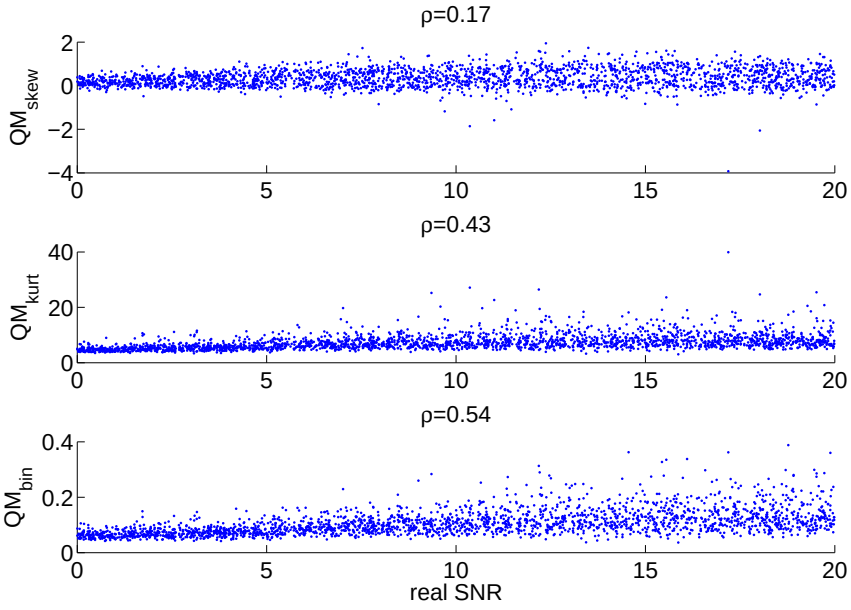


Figure 5.7 — Correlation between higher order statistics measures and real signal-to-noise ratio on a noisy version of the evaluation subset of XM2VTS. Each data point corresponds to an utterance.

Numerical performance evaluation of modality-specific quality measures

For the BANCA data, the quality measures based on VAD seem the most promising, as they exhibit correlation and normalised conditional mutual information with the score of client accesses. Out of the higher-order statistics based quality measures, kurtosis is the worst performer; the high Kullbak-Leibler divergence it exhibits can be attributed to a distribution with many outliers. Furthermore, higher-order statistics seem not to be useful in indicating classifier errors, as correlations are neglibigle.

For the XM2VTS data, the trend is reversed and in general higher-order statistics are more correlated with client scores. However, errors exhibit insignificant correlations and mutual information with all quality measures. This tends to indicate that XM2VTS is acquired in cleaner conditions than BANCA.

Accordingly, experiments on the noisy XM2VTS database (Table 5.7) show that errors (for clients) can more readlily be attributed to noise since correlation are higher than in the clean case. Skewness seems to be a poor indicator of signal quality.

For all databases, the effect mentioned in Section 5.4.4 is clearly visible, as correlation and mutual information is generally smaller for clients than for impostors. This indicates a need for caution in building joint models of scores or errors and quality measures.

measure	$\rho_{Sc\|\omega_0}$	$\rho_{Sc\|\omega_1}$	$I_{Sc\|\omega_0}$	$I_{Sc\|\omega_1}$	$\rho_{DR\|\omega_0}$	$\rho_{DR\|\omega_1}$	$I_{DR\|\omega_0}$	$I_{DR\|\omega_1}$	D_{KL}
QM_{VAD_E}	-0.18	0.49	0.04	0.15	0.28	0.11	0.08	0.04	0.93
QM_{VAD_H}	-0.17	0.48	0.04	0.12	0.27	0.09	0.07	0.04	1.22
QM_{skew}	-0.06	0.31	0.03	0.09	0.05	0.04	0.06	0.01	1.14
QM_{kurt}	-0.04	0.16	0.03	0.08	-0.02	0.07	0.09	0.01	7.45
QM_{bin}	-0.11	0.27	0.04	0.06	0.10	0.06	0.06	0.02	0.71

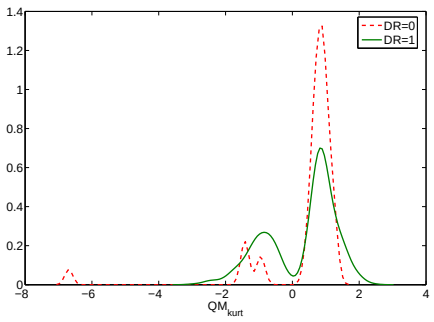
Table 5.5 — Average performance of modality-specific quality measures on the BANCA dataset.

measure	$\rho_{Sc\|\omega_0}$	$\rho_{Sc\|\omega_1}$	$I_{Sc\|\omega_0}$	$I_{Sc\|\omega_1}$	$\rho_{DR\|\omega_0}$	$\rho_{DR\|\omega_1}$	$I_{DR\|\omega_0}$	$I_{DR\|\omega_1}$	D_{KL}
QM_{VAD_E}	0.09	0.34	0.03	0.06	-0.05	0.15	0.01	0.03	0.08
QM_{VAD_H}	0.02	0.27	0.03	0.05	-0.03	0.11	0.01	0.02	0.12
QM_{skew}	0.08	0.52	0.03	0.14	-0.03	0.14	0.01	0.04	0.25
QM_{kurt}	0.07	0.43	0.03	0.09	-0.03	0.11	0.01	0.02	0.12
QM_{bin}	0.09	0.41	0.02	0.08	-0.05	0.10	0.01	0.02	0.02

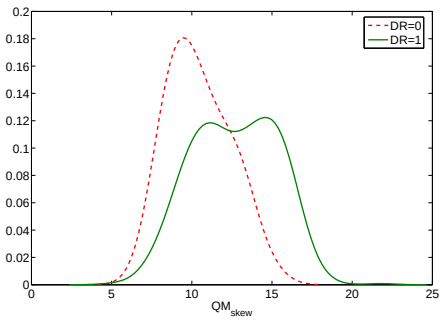
Table 5.6 — Performance of modality-specific quality measures on the XM2VTS evaluation dataset.

measure	$\rho_{Sc\|\omega_0}$	$\rho_{Sc\|\omega_1}$	$I_{Sc\|\omega_0}$	$I_{Sc\|\omega_1}$	$\rho_{DR\|\omega_0}$	$\rho_{DR\|\omega_1}$	$I_{DR\|\omega_0}$	$I_{DR\|\omega_1}$	D_{KL}
QM_{VAD_E}	-0.09	0.61	0.01	0.16	0.02	0.54	0.00	0.14	0.23
QM_{VAD_H}	-0.10	0.49	0.01	0.10	0.03	0.40	0.00	0.08	0.25
QM_{skew}	-0.01	0.01	0.00	0.04	0.01	0.02	0.00	0.02	0.28
QM_{kurt}	-0.03	0.29	0.00	0.07	0.02	0.23	0.00	0.05	0.26
QM_{bin}	-0.06	0.36	0.01	0.07	0.02	0.30	0.00	0.05	0.01

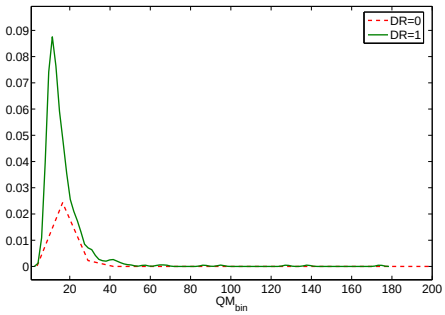
Table 5.7 — Performance of modality-specific quality measures on the XM2VTS noisy evaluation dataset.



(a) Kurtosis distribution.



(b) Skewness distribution.



(c) Centre bin measure distribution.

Figure 5.8 — Distributions of three quality measures based on higher-order statistics for correct (DR=1) and erroneous (DR=0) classifier decisions on BANCA G1 data.

5.8 Summary

In this Chapter, we have proposed a categorisation of classifier errors and quality measures. Modality-specific mesures are those which depend directly on the signal, while modality-independent measures do not, but are tied to a particular classifier. Introducing the concept of modality-independent quality measure, we have proposed two related measures of user model quality for probabilistic models, based on properties of the covariance matrix.

Depending on their intended use (error prediction or fusion), these quality measures can be evaluated in several ways. We have pointed out the deficiencies of assuming linear and homoscedastic distributions of scores: Normalised mutual information was proposed for evaluation of quality measures.

We showed that, unintuitively, the influence of noise can be different on client and impostor scores, and thus motivated the need for class-specific evaluation of quality measures. The normalised conditional mutual information thus becomes a useful tool in evaluating the class-specific effect of the quantity measured by quality measures.

The evaluation methods presented can serve as the basis for selecting quality measures when designing quality-dependent algorithms in biometric authentication. The practitioner should however be mindful that, as is the case in feature selection, the best proof of usefulness is obtained by using real classifiers.

In speech, we proposed several quality measures, based on segmentation in the time-domain, and based on higher-order statistics, and showed their good correlation with real signal-to-noise ratio on an artificially corrupted speech database.

Evaluation of quality measures on reference databases showed that both modality-specific and modality-independent quality measures contain information about classifier output scores, thus motivating the quest for quality-based algorithms in biometrics.

Reliability estimation in single-classifier verification

6

> I think that we had better correct an error into which we seem to have fallen in the use of the words "friend" and "enemy". What was the error, Polemarchus? I asked. We assumed that he is a friend who seems to be or who is thought good. And how is the error to be corrected? We should rather say that he is a friend who is, as well as seems, good.
>
> Plato, *The Republic*

6.1 Introduction

After having identified indicators of potential causes of variability (quality measures), we aim to model the relationship between theses causes of variability and the classifier's behaviour. Once a verification result has been obtained from a biometric verification classifier, it is often of practical interest to be able to measure the reliability* of the decision. The *reliability of a classifier's decision* can be phrased as "the probability of the classifier having taken a correct classification decision given available evidence". Another definition from Kukar and Groselj [169] is "[...] an estimated probability that the (single) classification is in fact the correct one".

The main differences between the "confidence measure" approaches we reviewed in Section 2.4 and our "reliability" method are in the domain of the evidence we use and the modelling approach. We define the estimation of reliability as an interpretable **probabilistic** method providing an output in the form of a posterior probability, based on combining **score modelling** and **quality measure modelling**. In this approach, quality measures are considered as additional features, and are essential for reliability modelling.

*In this thesis, "reliability" does not refer to the study of the life cycles of engineering products, but rather is used in the same as sense as that proposed by Toyama and Horvitz [305].

In Section 6.2, we propose a Bayesian network topology for inferring the reliability of a base classifier's decision. Section 6.3 then gives possible uses for reliability measures in biometric authentication. Section 6.4 talks about the specificities of evaluating reliability and confidence measures. Section 6.5 provides experimental results, and Section 6.6 concludes the Chapter.

6.2 A Bayesian network model of classification reliability

In order to represent the real state of the user identity and the verification result we introduce two binary variables: the class variable (Ω) and Classified user IDentity (CID). $\Omega = 1$ represents the event "the system user is a client", while $\Omega = 0$ corresponds to the event "the system user is an impostor". $CID = \{0, 1\}$ corresponds to the events "the biometric verification classifier accepts the identity claim" ($CID = 1$) and "the biometric verification classifier rejects the identity claim" ($CID = 0$). To define the reliability measure we introduce another binary variable DR, where $DR = 1$ represents that the "decision is reliable" or "the classifier is correct" and $DR = 0$ represents the opposite statement. In Boolean logic terms, $DR = \overline{CID \oplus \Omega}$.

The Bayesian network in Fig. 6.1 depicts an influence model for the variables Ω,CID and DR. In this network the True user IDentity (Ω) can be seen as the cause of a particular Classified user IDentity (CID) value. Indeed, a classifier performing above chance is more likely to accept identity claims if the biometric presentation truly originates from a client ($\Omega = 1$) than from an impostor ($\Omega = 0$). The Decision Reliability (DR) variable can be seen as an alternative source of errors in the CID value. It is there to summarise the influence of decision errors not strictly related to the user identity, such as intrinsic classifier performance and signal condition.

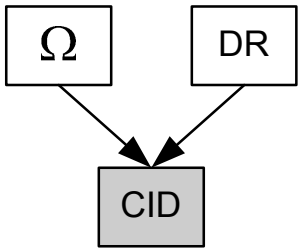


Figure 6.1 — Bayesian network for estimation of decision reliability

In this case, the set of nodes $V = (\Omega, CID, DR)$, and taking into account the arcs defined in Fig. 6.2, the joint pdf over V can be written as:

$$P(\Omega, CID, DR) = P(\Omega)P(DR)P(CID|DR, \Omega) \tag{6.1}$$

Since the variables Ω and DR are not observable during biometric verification, this approach would simply yield a confidence value equal to the maximum likelihood value learned during training, thus the reliability of the decision ($P(DR = 1|CID)$) would always be proportional to $(1 - err_{\omega_0|\omega_1})$, where $err_{\omega_0|\omega_1}$ is the error rate for impostors or clients on the training set.

6.2.1 Observable evidence for reliability estimation

Thus, we need to provide additional sources of information that can be observed and can provide evidence in favour of particular (Ω, DR) values. The verification score output from the base classifier carries information about the state of the user identity (client/impostor). We define the continuous

random variable Sc, corresponding to the verification score (measurement-level classifier output), and add it to the model as shown in Figure 6.2.

Quality measures measure can be used to provide evidence for the DR variable. For example, in speaker verification, the signal-to-noise (SNR) ratio of the speech signal can be used to measure the level of the acoustic noise. Therefore, we define an additional random variable **QM**, corresponding to the a vector of Quality Measures for the given modality (for example SNR estimated from an entropy-based speech-pause segmentation and a kurtosis-based measure of noise in the case of speech, see Chapter 5). The Bayesian network corresponding to this modification is shown in Fig. 6.2. It is interesting to note that running a structure learning algorithm such as K2 [1] with Bayesian or BDEu scoring on the above mentioned random variables with a real dataset results in a topology that is very similar to the one proposed here.

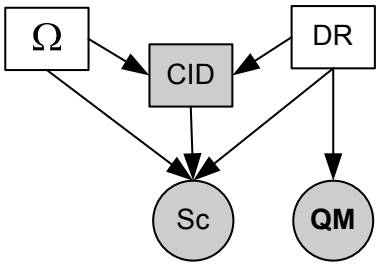


Figure 6.2 — Bayesian network with evidence variables for estimation of decision reliability

DR and Ω can be seen as "causes" for the observed Sc value, CID as a "consequence" of Sc, and DR is taken as a discrete node partitioning the continuous range of QM values. However, the semantic of the arcs is not taken to mean strict causality*, but as dependence or correlation. This is because we are mostly interested in the distribution over the nodes. This is justified by the observation that a joint distribution $P(A, B)$ can be factored either as $P(A|B)P(B)$ or as $P(B|A)P(A)$. A causal interpretation of the first factorisation might lead one to believe that "B causes A", while the second might engender "A causes B". Since both represent the same distribution, causation is merely an effect of interpretation.

Taking into account the evidence variables, the set of nodes $V = (\Omega, CID, DR, Sc, \mathbf{QM})$, and with respect to the arcs defined in Fig. 6.2, the joint pdf over V can be written as:

$$\begin{aligned} P(\Omega, CID, DR, Sc, \mathbf{QM}) = P(\Omega)P(DR)P(CID|DR, \Omega) \\ \cdot P(Sc|\Omega, DR, CID)P(\mathbf{QM}|DR) \end{aligned} \tag{6.2}$$

Since by definition not all combinations of Ω, DR, CID are valid (for instance $\Omega = 0, CID = 0, DR = 0$ is not possible), the $P(Sc|\Omega, DR, CID)$ term (the score term) represents four, not eight, distinct Gaussian distributions. These are the distributions of scores in case of correct reject (CR, $\Omega = 0, CID = 0$), correct accept (CA, $\Omega = 1, CID = 1$), false reject (FR, $\Omega = 1, CID = 0$), and false accept (FA, $\Omega = 0, CID = 1$). These four distributions are depicted in idealised form in Fig. 6.3.

The quality measure term in Eq.(6.2), $P(\mathbf{QM}|DR)$, represents two different Gaussian distributions, one for unreliable decisions ($DR = 0$), and one for reliable decisions ($DR = 1$).

A similar architecture has been used by Toyama and Horvitz [305] for a head tracking application in computer vision. In their case, several visual tracking algorithms are combined and the reliability

*a difficult concept to define in static Bayesian networks where there is no notion of time, and therefore no "before" or "after"

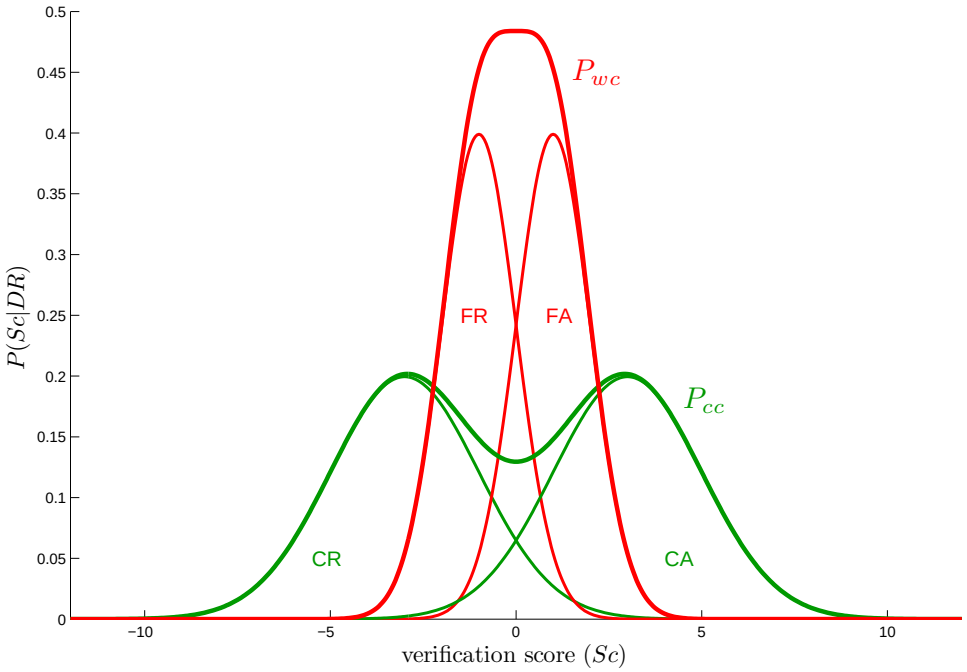


Figure 6.3 — (repeated from Figure 2.4.2) Idealised graph of correct verification ($P_{cc}(Sc)$) and verification error ($P_{wc}(Sc)$) score distributions showing the four sub-distributions: correct reject (CR), false reject (FR), false accept (FA), and correct accept (CA). Note that in reality the sub-distributions are likely to be non-Gaussian and overlap in a different way.

of each is estimated before taking a final decision. While the core of the rationale for the basic network topology is the same, there are several differences between the two. The first is that their network uses only discrete variables, which is not appropriate in our problem setting since we want to deal with continuous-valued scores and quality measures. The second is that the goal of inference in their case is the class variable Ω, while ours is the variable DR. The third is that the DR node is hidden in their case, and must be inferred from data, while we explicitly label DR as a binary variable during training, and define its meaning clearly. Fourthly, the quality measures are modelled using a naïve Bayes topology, so correlations are learned by the distribution of DR – since it is hidden, the quality measures are not independent by virtue of d-separation rules. In our case, the quality measures are modelled using a vector-valued node, meaning the correlations between quality measures can be explicitely learned. Lastly, their classifier decision variable being discrete, there is no "soft evidence" in the form of scores, which in our case would incur a considerable loss of information.

One problem with the Bayesian network defined in Fig. 6.2 is that the four score sub-distributions (FA, FR, CA, CR) are modelled as Gaussians, an assumption which gives reasonable practical results (see Section 6.5) but can be far from the score distributions available with today's limited biometric databases. An exemple of real score distributions is shown on Fig. 6.4

Likewise, in the network defined in Fig. 6.2, the distribution of quality measures is assumed Gaussian, an assumption which is not supported by the experimental data extracted from many different databases. An example of this is shown for the speech modality in Fig. 6.5, where the distribution of quality measures with respect to reliable and unreliable decisions is clearly non-

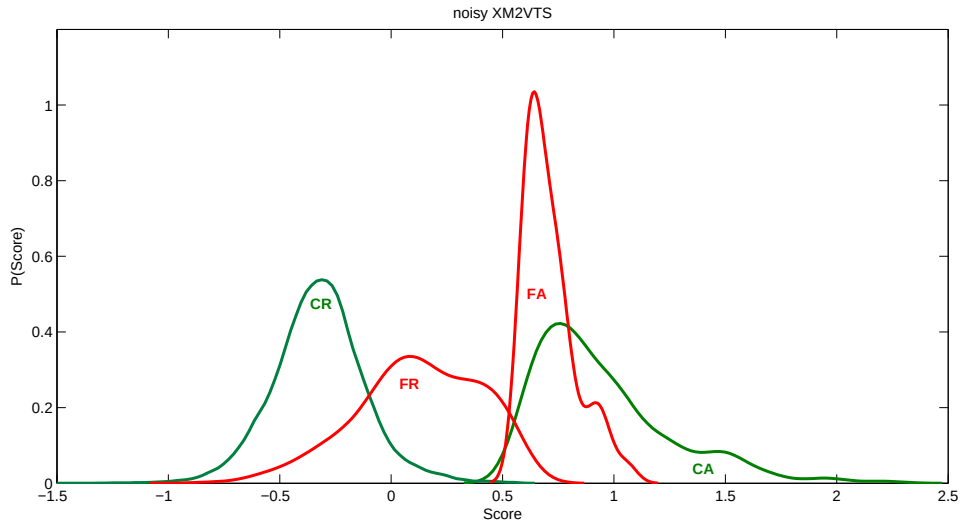


Figure 6.4 — Graph showing the four score sub-distributions: correct reject (CR), false reject (FR), false accept (FA), and correct accept (CA) for a speech classifier on a noisy version of the XM2VTS database. Note that the relative probability mass of the sub-distributions is not taken into account in order to show the density shapes more clearly. Contrast with the idealised version in Fig. 6.3

Gaussian.

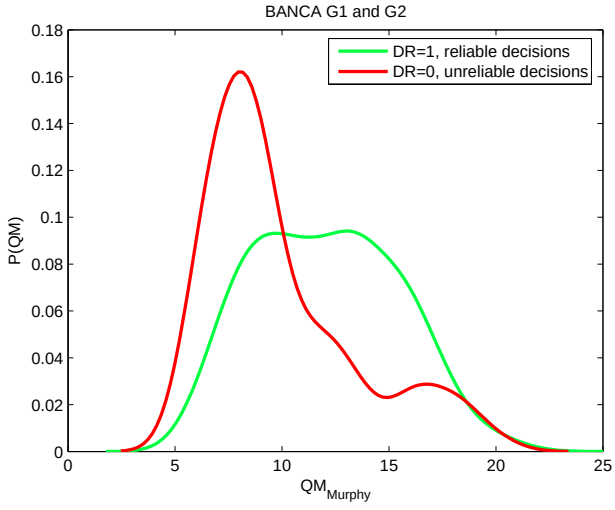


Figure 6.5 — Distributions of a quality measure on BANCA (group 1 and group 2) for reliable and unreliable classifier decisions. The QM_{Murphy} quality measure is explained in Chapter 5.

Therefore, the Bayesian network used to model reliability is further refined to allow for more complex distributions of score and quality measures.

6.2.2 Modelling non-normal evidence

A family of distributions which is flexible enough to model any other distribution is that of Gaussian mixture models [190]. As shown in Section 4.3, there exists a topology for Bayesian networks which forms a Gaussian mixture model if the continuous nodes are Gaussian and the mixing coefficients are represented by discrete nodes. This modification of the reliability model is shown on Fig. 6.6.

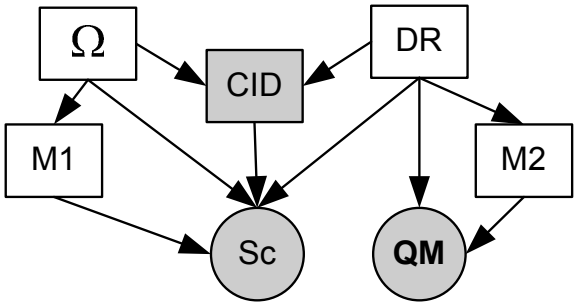


Figure 6.6 — Bayesian network model with mixture modelling of evidence nodes for estimating reliability

Given the Bayesian network variables set $V = (DR, \Omega, CID, Sc, \mathbf{QM}, M1, M2)$, where $M1, M2$ represent mixing weights learned through a maximum likelihood algorithm, and taking into account the arcs defined in the Bayesian network of Fig. 6.6, the joint pdf over V can be factored as follows:

$$\begin{aligned} P(V) = P(DR)P(\Omega)P(CID|\Omega, DR)P(Sc|DR, \Omega, CID, M1) \\ \cdot P(M1|\Omega)P(M2|DR)P(\mathbf{QM}|DR, M2) \end{aligned} \tag{6.3}$$

The posterior $P(DR|CID, Sc, \mathbf{QM})$ is the distribution of the decision reliability measure. Following the network topology in Fig. 6.6 the posterior distribution over DR can be written as:

$$P(DR|cid, sc, qm) = \alpha \sum_{\Omega, M1, M2} P(V) \tag{6.4}$$

For a given value of DR, say $DR = 1$, and a classifier decision $CID = cid$, the distributive law can be applied to Eq. 6.4 to simplify the computation:

$$\begin{aligned} P(DR = 1|cid, sc, \mathbf{qm}) \quad = \quad & \alpha P(DR = 1)P(\Omega) \\ & \cdot \sum_{M1} \underbrace{P(M1|\Omega)}_{\star mixing\ coefficient} \cdot P(sc|DR = 1, M1) \\ & \cdot \sum_{M2} \underbrace{P(M2|DR = 1)}_{\star mixing\ coefficient} P(\mathbf{qm}|DR = 1, M2), \end{aligned} \tag{6.5}$$

where α is a normalisation coefficient:

$$\alpha = \frac{1}{P(cid, sc, \mathbf{qm})} \tag{6.6}$$

The term $P(\Omega)$ is the prior probability on client or impostor access happening. This term can be set depending on the application (see Section 6.2.6). The $P(DR = 1)$ term is the prior probability of the classifier decision being correct. The terms marked with $\star$ effectively act as mixing coefficients, and the term within the $M2$ summation corresponds to a two-components Gaussian mixture model over the quality measures. The $P(Sc|DR, \Omega, CID, M1)$ term in Eq. 6.3 can be simplified to $P(sc|DR = 1, M1)$ in Eq. 6.5, because a value of 1 for DR means by definition that

$\Omega = CID$, and a certain classifier decision, $CID = cid$ will then be reflected by $\Omega = cid$. In this case, the term defines a single Gaussian distribution.

Thus, a 2-component Gaussian mixture model is used to model the distribution of scores in the cases of CR, FA, FR, and CA. This essentially decomposes the two classical client ($P(Sc|\Omega = 1)$) and impostor ($P(Sc|\Omega = 0)$) distributions in four sub-distributions, each having the possibility of deviating from the Gaussian distribution.

6.2.3 Quality-measure specific topology refinements

While the above topology is generic enough to give good modelling accuracy with many quality measures and modalities (it has been used with speech, face, and signature), it is possible to tailor it to specific quality measures in order to better model the dependencies between the quality measure and other variables. As this is a data-dependent process, we can use the indicators developed in Section 5.4 to provide evidence in favour of specific topologies.

One such refinement is mandated in some cases because the quality measure has a different relationship with client scores than with impostor scores (see Section 5.4.4). Accordingly, we can make the distribution of quality measures dependent upon the value of the Ω node by adding an edge $\Omega \rightarrow \mathbf{QM}$. This results in replacing the $P(\mathbf{QM}|DR, M2)$ term in Equation (6.3) by $P(\mathbf{QM}|\Omega, DR, M2)$.

Also, in the cases where the dependency between scores and quality measures is deemed important enough (say, above a particular correlation or mutual information threshold), it is possible to add an edge $\mathbf{QM} \rightarrow Sc$.

6.2.4 Influence of signal quality on the reliability posterior

Signal quality, as all the quality measures that can be used in the reliability framework (see Chapter 5), changes the shape of the reliability posterior. Higher quality signals should lead to a more reliable decision. In terms of the reliability posterior, and depending on the overlap of the base classifier's score distributions, this translates to either a larger part of the score range having a high reliability value, or a shift in the score-space boundary between reliable and unreliable decisions.

In order to illustrate this point and to enable an intuitive comparison with the confidence measures presented in Section 2.4, Figure 6.7 presents graphs of the $P(DR = 1|CID, Sc, \mathbf{QM})$ posterior with various values for the quality measure, computed using the Gaussian assumption reliability model of Section 6.2.1. The reason why the posteriors look different in the client and impostor cases is because the underlying distribution of scores have different overlaps for clients than for impostors.

6.2.5 Parameter estimation for single-classifier reliability

In training, all nodes shown in Fig. 6.6 save $M1$ and $M2$ are observed (visible). As depicted in Fig. 6.8, the observations used to estimate the conditional densities for nodes $\Omega, CID, DR, Sc, \mathbf{QM}$ come from running a single-modality classifier (for instance a face verification system) on a development dataset. The development dataset should contain biometric presentations taken in environmental conditions comparable with that of deployment, so the range of variability can be modelled.

All visible nodes are learned through maximum likelihood over the development set, and the hidden variables (mixing weights $M1$ and $M2$) are learned using Expectation-Maximisation as exposed in Section 3.3.

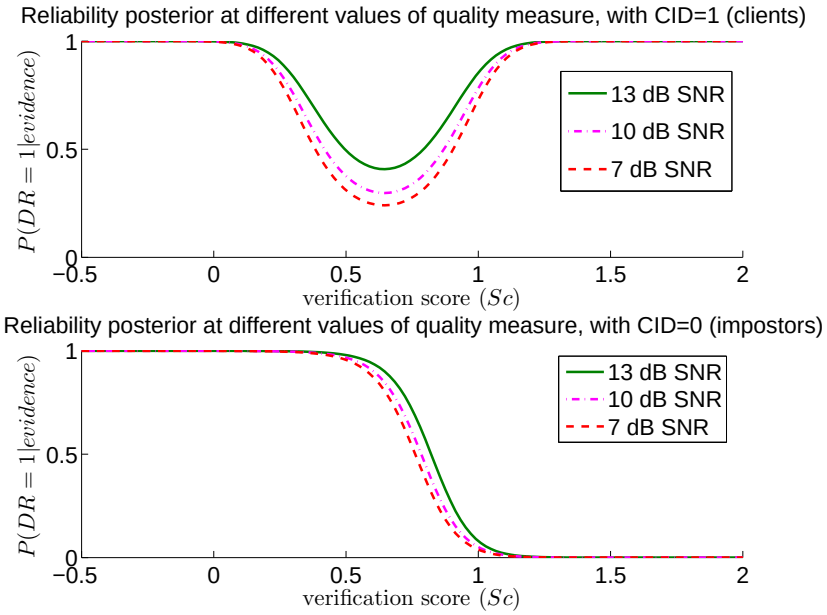


Figure 6.7 — Example reliability posterior $P(DR = 1|QM, Sc, CID)$ at various levels of acoustic noise on the XM2VTS database

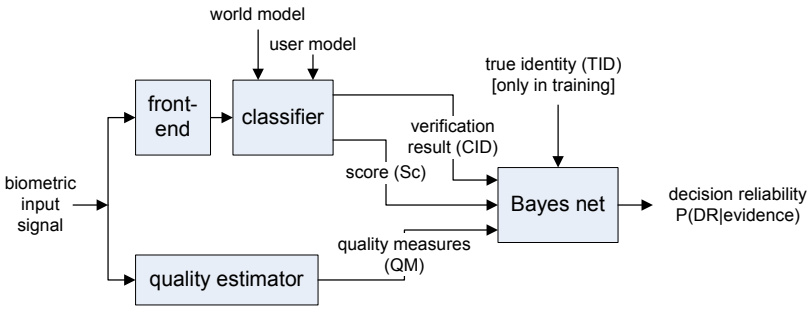


Figure 6.8 — Combined single-modality verification system and Bayesian network for reliability estimation

6.2.6 Setting priors for reliability models

The prior on the classifier error $P(DR)$ (prior probability of the classifier decision being correct or not) and the prior on the user identity $P(\Omega)$ (prior probability on client or impostor access) have special importance. Many confidence measures (see Section 2.4) do not acknowledge explicitely the existence of these priors, and so do not allow for their flexible setting. Having them as explicit model parameters allows for simple modification of the model without the need for retraining the rest of the model.

Both priors can be set in several ways. The first approach is to set them as a uniform prior $(0.5, 0.5)$ if no information is available or there is a need to balance type I and type II errors. The second approach is to set them "manually" if expert knowledge is available, or if a particular trade-off in error rates is desired. The third way is to learn these priors on training data, an approach which makes the assumption that the data used during deployment of the system will correspond to the training set.

The $P(\Omega)$ prior reflects the expected amount of impostors in the population. As mentioned in

Section 2.7.4, in biometric testing databases, the amount of impostor data is generally significantly larger than the amount of client data. This client-impostor class imbalance is likely to be reversed in real deployments, such as identity documents applications, where it can be assumed that most users will be genuine clients and a vanishingly small minority will be impostors. Therefore, it is crucial that this prior be easy to set according to expert opinion on likely class distribution.

The $P(DR)$ prior is a probability of the classifier taking a correct ($P(DR = 1)$) or incorrect ($P(DR = 0) = 1 - P(DR = 1)$) decision. Since this prior is classifier-dependent, it should be learned on a database containing data that is representative of the deployment conditions. If the testing conditions are not entirely known in advance, as is often the case, this can be fixed at 0.5.

6.3 Uses of reliability models

6.3.1 Using the reliability model to elicit a posterior probability of client identity

In this section we would like to offer an alternative interpretation and use of the model presented in Section 6.2, where we focused on obtaining estimates for the probability that the classifier has made a mistake.

Trained classifiers using estimation of probability densities can in general be used to produce estimates for the posterior $P(\Omega = \omega_i|\mathbf{O})$, meaning the probability that, given the observation vectors O, the correct class for the object under scrutiny is indeed ω_i. Another interpretation for this quantity is *the probability that the assigned label ω_i is correct*, which can be called the *confidence in the classification result* [75]. As mentioned in Section 2.4, many authors take the stance that posterior probabilities can serve as confidence measures, an approach which has been applied widely in pattern recognition.

In probabilistic classifiers such as Gaussian mixture models, Bayesian networks, or hidden Markov models, the posterior can be directly interpreted as a confidence measure by transforming and normalising the likelihood output by the classifier via Bayes rule. Others, like multilayer perceptrons, give output which can be considered a posterior probability*. For a large number of other classifiers (for instance k-nearest-neighbours), we cannot simply relate the posterior to a probability density: there is no density, only an output quantity indicating "similarity" between class and input vectors. In these cases, it is necessary to map the output of the classifier to behave like a probability density and obey the basic axioms of probability. This can for instance be achieved by using logistic regression [75], or custom transformations devised specifically for each classifier type.

The reliability approach uses a Bayesian network to model the output of the classifier (the measurement level), be it a log-likelihood ratio, an Euclidean distance or another type of similarity measurement, as a Gaussian mixture model. Thus, we largely abstract over the problem of devising a specific transformation for each type of classifier: we only assume that the ouput of many classifiers on two-class problems can be modeled with enough flexibility by a mixture of Gaussians, and that the measurements output by classifiers will generally cluster into erroneous and correct classifications.

The Bayesian network presented in Fig. 6.6 can be used to perform inference directly on the Ω node, resulting in the conditional posterior $P(\Omega = \omega|CID, Sc, \mathbf{QM})$. This posterior, in turn, can be considered as a confidence measure in the classification result. Applying a threshold (typically 0.5 to obtain the *maximum a posteriori* decision) on this posterior allows for reliability-based inference of client identity.

*Some authors disagree with this interpretation [180].

From the joint probability expressed in Eq. (6.3), we can write the posterior of interest as:

$$P(\Omega|cid, sc, \mathbf{qm}) = \alpha \sum_{DR,M1,M2} P(DR, \Omega, CID = cid, Sc = sc, \mathbf{QM} = \mathbf{qm}, M1, M2), \tag{6.7}$$

where the α term is as per Eq. (6.6).

6.3.2 Classification with the reject option

Many decision errors in biometric verification are due to ergonomic factors rather than algorithmic weaknesses. For instance, in iris and face verification improper distance and centering of the image can significantly degrade verification accuracy. In speech-based verification, distance from the microphone and speaking volume are important factors. In this section, we propose a strategy to cope with uncertain classification results by re-acquiring the signal up to N times. This idea is present in other fields of pattern recognition such as optical character recognition, where an example is that the second-stage recogniser can reject the character segmentation proposed by the preprocessing module if the confidence value associated to it is too low [49].

For example, in an interactive speaker verification system, the user could be asked to move closer to the microphone if the signal-to-noise-ratio is too low, or the operator could be informed that verification results for presentation n are unreliable. In this case, performing sequential repair only if needed presents the advantage of minimising the amount of interaction between the user and the system, thus speeding up the verification process. The final classifier decision $FCID$ can then be presented as a definitive verification result.

The sequential repair strategy outlined in Fig. 6.9 is equivalent to doing single-classifier fusion of repeated acquisitions with binary weights at the score level, where the score of the unreliable presentation(s) gets a weight of 0 and the reliable presentation gets a weight of 1. Instead of throwing away all of the information provided by the first presentation, it is possible to combine it with the second presentation (see for instance [150] for this strategy applied to the face modality). A simple scheme is to weight each presentation score by its corresponding normalised reliability value to derive the final (fused) score:

$$Sc = \sum_n rel(Sc_n) \cdot Sc_n, \tag{6.8}$$

where the normalised reliability values are obtained in the following fashion:

$$rel(Sc_n) = \frac{P(DR = 1|CID_n, Sc_n, QM_n)}{\sum_n P(DR = 1|CID_n, Sc_n, QM_n)} \tag{6.9}$$

In this case, the decision to acquire a new presentation would still be governed by the insufficient reliability of the first presentation. The advantage of this scheme over a scheme that would always acquire two presentations is that the interaction time with the biometric verification system can be minimised. By setting the reliability threshold, it is possible to bias the system towards being more tolerant of low reliabilities (resulting in higher error rates), or less tolerant (resulting in longer interaction time with the system for users).

We have shown empirically the effectiveness of a reliability-based sequential repair scheme in [254, 260, 261].

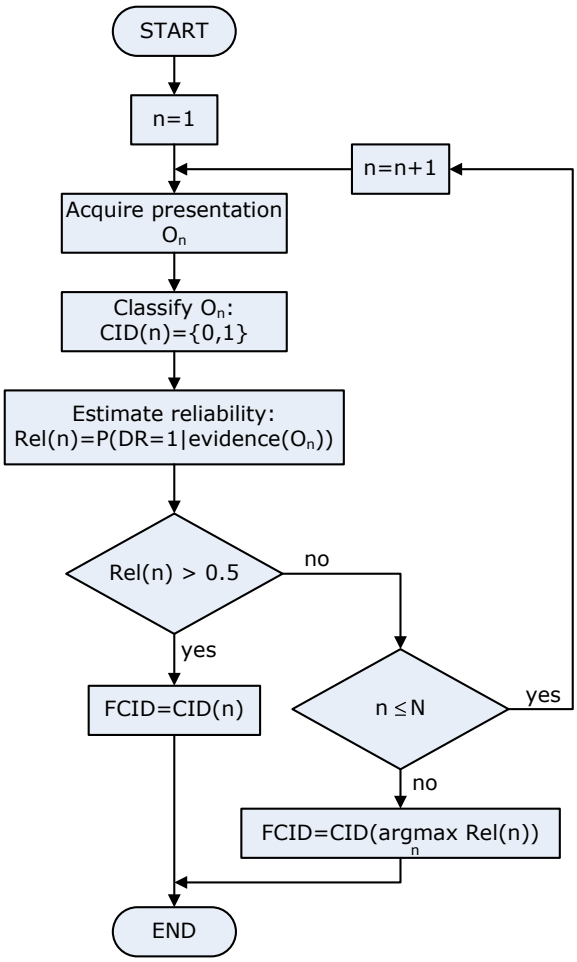


Figure 6.9 — Sequential repair algorithm based on reliability estimation

6.3.3 Using reliability for decision correction

The reliability model can be used to estimate, on an instance-by-instance basis, when the decision of the base classifier is likely to be unreliable. In such cases, the decision can be "rigged" by inverting it. Kryszczuk and Drygajlo [164] have explored the role of prior probabilities in the inversion process.

Denoting the base classifier decision by a binary variable CID (0 for impostors, 1 for clients), the reliability classification by a binary variable DR (0 for unreliable, 1 for reliable), and the rigged decision by RD, the decision rigging works by implementing the negative exclusive-or function: $RD = \overline{CID \oplus DR}$.

This use of single-classifier reliability can be applied to the combination of classifiers in the multi-classifier context, as shown in Section 8.5.

6.4 Evaluation of reliability models

The first measure of performance that we use for assessing reliability and confidence measures is the accuracy of prediction of decision correctness. In most biometric databases, the number of samples per class* (clients, $\Omega = 1$, and impostors, $\Omega = 0$) is heavily imbalanced (up to 3 orders of magnitude for XM2VTS), hence we cannot take the classical definition of accuracy as $\frac{nCorrectClassifications}{nSamples}$, or the performance of the confidence and reliability measures for client accesses would have very little influence on the overall results. Furthermore, since the baseline classifier has an error rate of less than 50% (otherwise it should not be used), there will always be less cases where $DR = 0$ than cases where $DR = 1$. Thus, a blind confidence measure could predict $DR = 1$ all the time and be mostly correct if this imbalance is not accounted for. Since we have a "double imbalance" situation, we don't make use of the geometric mean which can be useful in "single imbalance" situations [16, 168], but rather we define balanced accuracy as

$$acc_{bal} = \frac{1}{4} \sum_{dr=\{0,1\}} \sum_{\omega=\{0,1\}} \frac{Ncorr_{DR=dr,\Omega=\omega}}{N_{DR=dr,\Omega=\omega}}, \tag{6.10}$$

where $Ncorr_{DR=dr,\Omega=\omega}$ is the number of correctly classified samples out of a total of $N_{DR=dr,\Omega=\omega}$ samples with ground truth labels $DR = dr$ and $\Omega = \omega$. This measure expresses the overall performance of the reliability or confidence measure. A measure that performs well for, say, impostors, but not for clients will thus be penalised by this evaluation criterion. As can be seen from Equation (6.10), the balanced accuracy is composed of four terms: acc_{CA} is the accuracy on correct accept cases ($\Omega = 1, DR = 1$), acc_{CR} is the accuracy on correct reject cases ($\Omega = 0, DR = 1$), acc_{FA} is the accuracy on false accept cases ($\Omega = 0, DR = 0$), and acc_{FR} is the accuracy on false reject cases ($\Omega = 1, DR = 0$).

The performance of confidence measures over a set of test data can also be evaluated by producing a DET curve (see Section 2.7.2) based on two distributions of confidence or reliability measures: one for the measures over correct decisions ($DR = 1$), and one for the measures over wrong decisions ($DR = 0$). The less overlap between the distributions there is, the better the confidence or reliability measure will be. DET curves are a meaningful tool to compare confidence and reliability measures only if these are trained with the same assumptions about the imbalance of the training set. In the present case, CM_{Gauss}, $CM_{Logistic}$ (with uniform priors on Ω), CM_{Margin} (with equal cost for false accept and false reject in building the FAR, FRR curves) and reliability (with uniform priors on Ω

*in the following discussions, *class* will mean impostor ($\Omega = 0$) or client ($\Omega = 1$) access when talking about the speaker verification classifier. When talking about the reliability or confidence measure, which can be considered as a second-level classifier, *class* will be taken to mean correct ($DR = 1$) or incorrect ($DR = 0$) decision.

and DR) can be compared, because the structure of the testing set in terms of Ω-DR class balance will have little impact on the results of the test.

CM_{Bayes} however is based on direct modelling of the correct and erroneous decisions score distributions (CA, CR, FA, and FR, see Figure 6.3) and thus will be favoured by a test set structure matching the training set structure (small data counts for CA, FR with respect to CR, FA). Therefore, direct comparison makes sense only in this scenario.

Another objective measure of goodness for reliability or confidence measures is normalised cross-entropy. It can be defined as the "relative decrease in uncertainty about the classifier's decision provided by the confidence measure", while the original definition from NIST for speech recognition confidence measures [204] is "the mutual information (cross entropy) between the correctness of the system's output word and the confidence score output for it, normalized by maximum cross entropy". However, this measure is also biased in favour of confidence or reliability estimates that perform better on the majority class ($DR = 1$). Thus, while it is very useful in speech recognition applications, we do not use it for evaluation in the current biometric identity verification setting given the imbalance of classes.

6.5 Experiments and results

In these experiments, we aim to empirically examine the performance of reliability measures compared to the performance of state-of-the-art confidence measures exposed in Section 2.4. Our criterion is the ability to predict whether a given classifier decision is correct or not, that is the class of interest in classification is DR, not Ω. We report performance in terms of DET curves and balanced accuracies.

6.5.1 Reliability in speaker verification

In these experiments, we use the speaker verification classifier described in Section 4.4.5. The quality measure used is QM_{VAD_E} (see 5.5.1).

Accuracy of reliability models and confidence measures

On BANCA (Figure 6.10, Figure 6.11, Table 6.1) the reliability measure outperforms other confidence measures in terms of balanced accuracy. Apart from better score modelling, the noisy environment of BANCA, as indicated by the quality measure, is one of the main reasons for the performance.

On XM2VTS (Figure 6.12, Table 6.2), the reliability measure also performs better than other confidence measures in terms of balanced accuracy, but is close to the CM_{Bayes} measure in terms of accuracy on DR prediction, as shown on the DET curve. The CM_{Bayes}, which detects correctly 0% of correct accepts, is not penalised much because of the clients-impostors imbalance over the dataset (three orders of magnitude). Correspondingly, all speech quality measures presented have non-existent correlation and mutual information with false accepts (see Table 5.7, thus the gains provided by better modelling of false rejects are not substantial.

These results confirm our earlier work on other databases [254, 260, 261].

6.5.2 Reliability in signature verification

The classifier used for these experiments is a BN/GMM with 2 Gaussian components, using 11 global features. The data is preprocessed by linear interpolation. The quality measure used is $\overline{QM}_{det_w}$

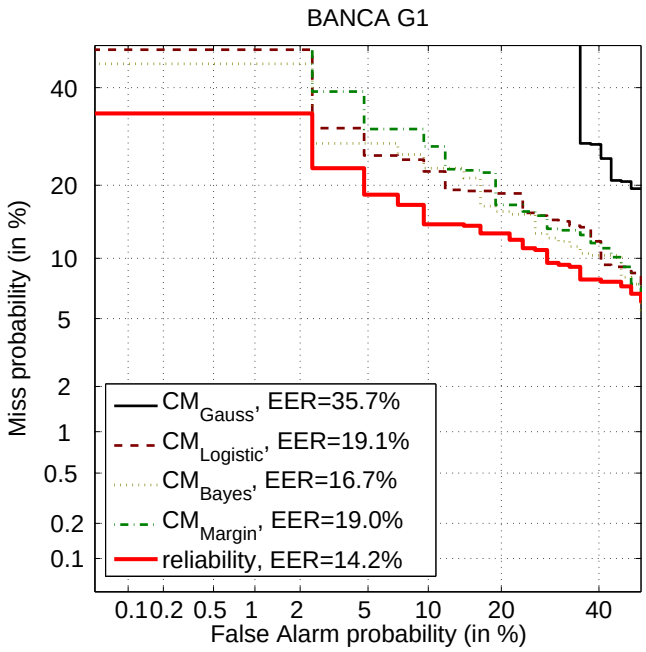


Figure 6.10 — Confidence and reliability experiments: results on BANCA G1.

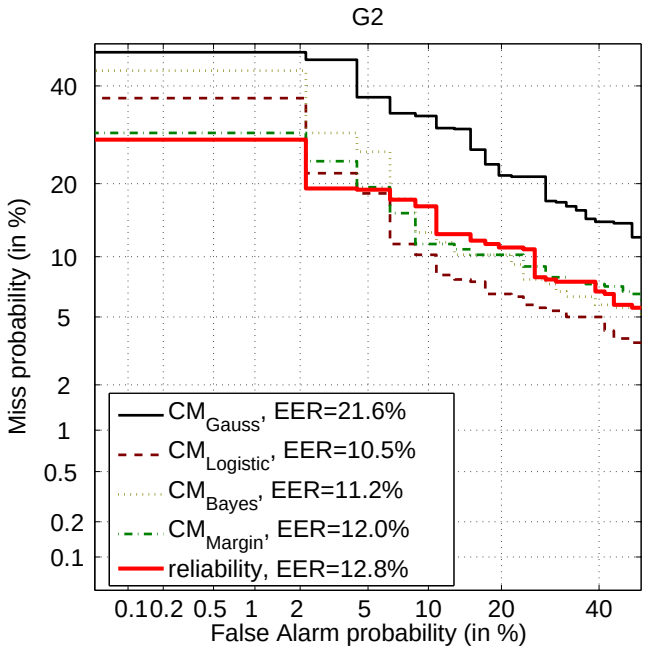


Figure 6.11 — Confidence and reliability experiments: results on BANCA G2.

method	acc_{CA} [%]	acc_{CR} [%]	acc_{FA} [%]	acc_{FR} [%]	acc_{bal} [%]
CM_{Gauss}	46.98	83.27	96.15	50.83	69.31
$CM_{Logistic}$	96.53	99.48	27.08	10.00	58.27
CM_{Bayes}	86.31	74.17	87.98	80.56	82.25
CM_{Margin}	53.11	54.33	100.00	97.22	76.17
Reliability	80.96	86.92	91.99	86.94	86.70

Table 6.1 — Decision correctness prediction for reliability and confidence measures on BANCA. All accuracies are averaged over G1 and G2 and given in percent.

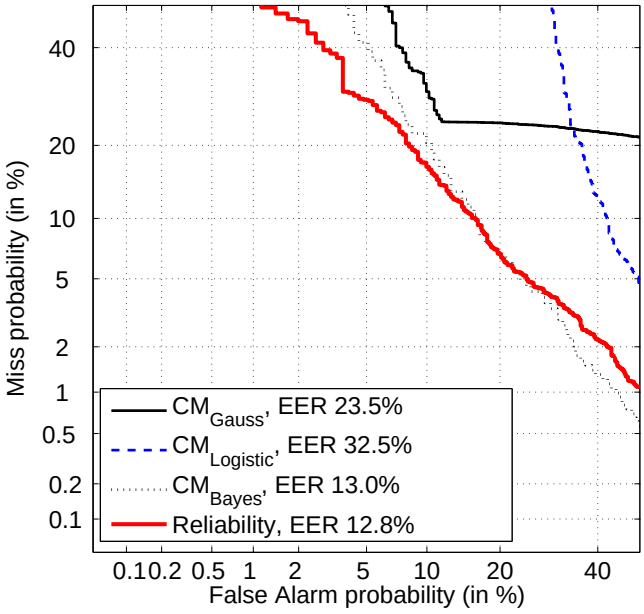


Figure 6.12 — Confidence and reliability experiments: results on the noisy version of XM2VTS. The results for CM_{Margin} are not shown since the EER is more than 50%.

method	acc_{CA} [%]	acc_{CR} [%]	acc_{FA} [%]	acc_{FR} [%]	acc_{bal} [%]
CM_{Gauss}	56.43	86.75	2.17	25.38	42.68
$CM_{Logistic}$	100.00	93.95	0.00	71.54	66.37
CM_{Bayes}	0.00	91.17	100.00	77.69	67.22
CM_{Margin}	100.00	45.30	0.00	74.23	54.88
Reliability	71.43	83.58	98.91	86.54	85.11

Table 6.2 — Decision correctness prediction for reliability and confidence measures on the noisy version of XM2VTS. All accuracies are given in percent.

(Section 5.6.2, Equation (5.39)).

Accuracy of reliability models and confidence measures

On BMEC (Figure 6.13, Table 6.3), the reliability measure outperforms other confidence measures, except for CM_{Margin}. The good peformance of CM_{Margin} can be explained by the client:impostor attempt ratio being not far from 1, and the kernel-based score modelling performed by this measure. Conversely, the lack of improvement due to the quality measure indicates that, at least for this classifier and quality measure, the topology of the reliability model may not be adequate. It may also be the result of the low dependence between this quality measures and errors made by this classifier, as indicated by DR.

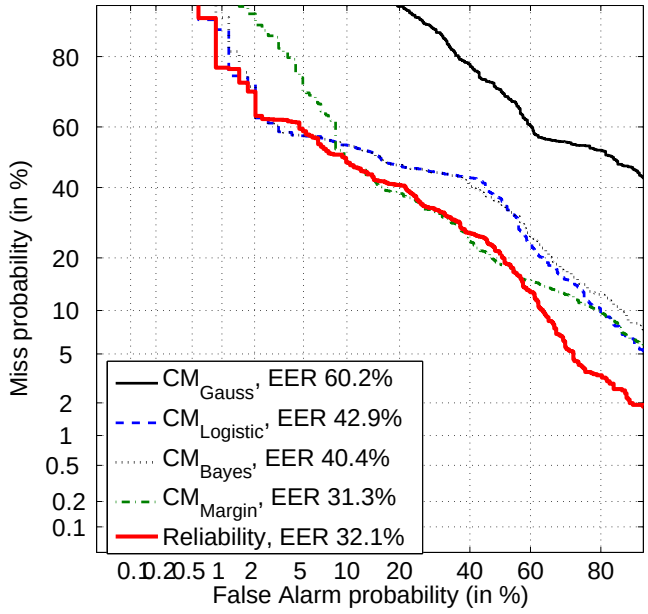


Figure 6.13 — Confidence and reliability experiments: results on the BMEC 2007 signature database. Note the scale of the graph is different than ordinary, to show more of the range.

method	acc_{CA} [%]	acc_{CR} [%]	acc_{FA} [%]	acc_{FR} [%]	acc_{bal} [%]
CM_{Gauss}	0.00	0.00	100.00	100.00	50.00
$CM_{Logistic}$	0.00	100.00	100.00	0.00	50.00
CM_{Bayes}	0.00	85.57	100.00	65.54	62.78
CM_{Margin}	46.18	78.23	71.94	81.08	69.36
Reliability	54.15	72.76	66.33	85.81	69.76

Table 6.3 — Decision correctness prediction for reliability and confidence measures on BMEC2007. All accuracies are given in percent.

6.6 Summary

In practical applications, it is often useful to be able to gauge how much trust should be put in a classifier's output. Because, except in the simplest cases, the relationship between errors, classifier decisions, and scores are hard to explain analytically, we propose to learn a probabilistic model of classifier errors, in the form of a Bayesian network.

The observable evidence to favour high or low reliability consists in a training set of classifier outputs and ground truth class. However, a measure of factors that contributed to potential errors is needed to improve the modelling. This is provided under the form of quality measures. The effect of various level of quality, as measured by the quality measure, is to change the form of the posterior distribution.

Since most score distributions and quality measure distributions are not Gaussian, it is important to allow for modelling of mixture densities. Many existing confidence measures make simplifying assumptions about the form of the distributions which do not hold in real data.

The output of reliability models can be used for many purposes, including human examination or automated post-processing. The usefulness of the output can be gauged by plotting DET curves, or by computing numerical performance measures such as balanced accuracy, which take into account the "double-imbalance" problem in confidence and reliability modelling: there is generally less client data than impostor data in the training set, and there are less errors than correct decisions.

Experiments show that reliability modelling outperforms or at least perform as well as state-of-the-art measures, while offering additional interpretability and flexibility in parameter setting.

Bayesian networks for combining multiple classifiers

7

7.1 Introduction

Combining the output of several classification algorithms on the same task generally brings about more decrease in error than tuning a single classification algorithm on the same task [151, 270]. This principle is especially important in multimodal biometric authentication, where the diversity of base classifiers in different modalities promises significant improvements in classification accuracy, and where the failure of a classifier in a specific modality can be compensated by classifiers in other modalities. Indeed, for large-scale applications of biometric authentication, it can be expected that multimodal ensembling techniques will become widespread as some fraction of the target population is likely to posess at least one unstable or unusable biometric trait.

As mentioned in Section 2.6, two main families of fusion methods exist: fixed rules, and trained rules, which require parameter estimation over a training set. In this chapter, we focus on trained rules, and specifically on the use of Bayesian networks to combine different classification algorithms.

We show that Bayesian networks offer a flexible and powerful probabilistic framework for combining multiple classifiers by recasting several combination methods as probabilistic, and proposing new fusion models. We start by *generic* topologies (Section 7.2), in the sense that they are applicable to both continuous and discrete variables, making them suitable for both score-level and decision-level fusion. We then propose Bayesian network topologies for decision-level fusion in Section 7.3. Section 7.4) presents score-level classifier combination, and proposes a structure learning algorithm for classifier combination called sparse regression fusion. Section 7.5 shows experimental results on a variety of modalities and databases, and Section 7.6 closes the Chapter.

7.2 Generic topologies for multiple classifier fusion with Bayesian networks

In this section, we review several "classic" topologies that have been used for classification with Bayesian networks and show how these can be applied to the multiple classifier combination problem. We call these models generic in the sense that they can be used for score-level and decision-level multiple classifier fusion.

7.2.1 Naïve Bayes

The naïve Bayes (NB) classifier [140, 175] treats all features as independent from each other given the class. This is equivalent to saying that if the class variable is visible, we have enough information to determine the probability density of each feature independently of the others. While this assumption is rarely met theoretically or in practice (multiple classifier outputs on the same classification tasks are not independent), classification results can be quite good. Interesting recent results on the theoretical reasons behind this fact are found in [170].

The naïve Bayes classifier, as the name suggests, is based on the use of Bayes' rule:

$$P(\Omega|X_1 = x_1, \ldots, X_L = x_L) = \alpha P(X_1 = x_1, \ldots, X_L = x_L|\Omega)P(\Omega), \tag{7.1}$$

where Ω is the class variable, the X_l are the components of the L-dimensional feature vector $\mathbf{X}$ (attributes), and α is a normalisation constant. Assuming (naïvely) that the attributes are independent from each over given the class, we can use the chain rule of probability and directly rewrite Eq. (7.1) as:

$$P(\Omega|X_1 = x_1, \ldots, X_L = x_L) = \alpha P(\Omega) \prod_{l=1}^{L} P(X_l = x_l|\Omega). \tag{7.2}$$

From (7.2) it is clear that, since the conditioning set of all conditional terms is the singleton ω, the topology induced is a star-like structure, with arcs going away from the class node ω towards each feature node X_l. As an illustration, we show an example naïve Bayesian network for doing score-level fusion of 4 classifiers on Fig. 7.1(a).

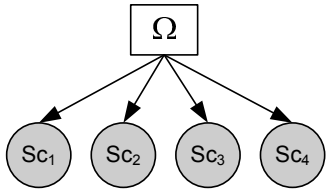
The naïve Bayes classifier can be used both with discrete (decision-level) and continuous (score-level) variables. For compactness in this section we will change variables and denote $C \stackrel{\triangle}{=} CID$.

Decision-level fusion

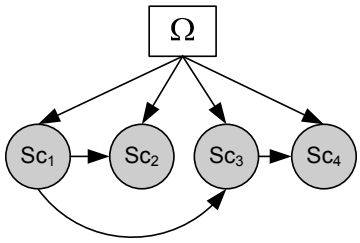
For discrete variables, the class-conditional probability mass functions are modelled as binomial distributions, giving:

$$\begin{aligned} P(C_l = c_l|\Omega = \omega) &= \frac{1!}{c_l!(1-c_l)!} p_{l\omega}^{c_l} (1-p_{l\omega})^{1-c_l} \\ &= p_{l\omega}^{c_l} (1-p_l)^{1-c_l}. \end{aligned} \tag{7.3}$$

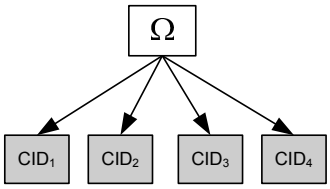
where $c_l \in \{0, 1\}$ is the lth classifier decision, and $p_{l\omega} \stackrel{\triangle}{=} P(C_l = c|\Omega = \omega)$ is the probability that ensemble classifier l has taken value c given that the class is t. This can be estimated by maximum likelihood on the development dataset. Since every density is estimated independently of each other, the number of "trials" (classifier outputs) in each binomial is 1 and the expression reduces to a Bernoulli distribution for each classifier $l = 1 \ldots L$. We thus term this decision-level



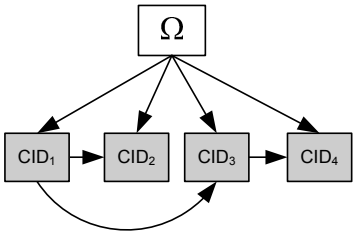
(a) Score-level fusion of 4 classifiers with naïve Bayes model



(b) Score-level fusion of 4 classifiers with example TAN model



(c) Decision-level fusion of 4 classifiers with naïve Bayes model



(d) Decision-level fusion of 4 classifiers with example TAN model

Figure 7.1 — Example score-level and decision-level multiple classifier fusion with naïve Bayes and TAN topologies.

combination method *Bernoulli fusion*. The posterior probability $P(\Omega|C_1, C_2, \ldots, C_L)$ is a product of Bernoulli distributions. For an example of fusing 3 classifiers, by applying Equation (7.2) we have:

$$\begin{aligned} P(\omega|c_1,c_2,c_3) &= \frac{P(\Omega=\omega)P(C_1=c_1|\Omega=\omega)P(C_2=c_2|\Omega=\omega)P(C_3=c_3|\Omega=\omega)}{\sum_\omega P(\Omega=\omega)P(C_1=c_1|\Omega=\omega)P(C_2=c_2|\Omega=\omega)P(C_3=c_3|\Omega=\omega)} \\ &= \frac{(p_\omega^\omega(1-p_\omega)^{1-\omega})(p_{1\omega}^{c_1}(1-p_{1\omega})^{1-c_1})(p_{2\omega}^{c_2}(1-p_{2\omega})^{1-c_2})(p_{3\omega}^{c_3}(1-p_{3\omega})^{1-c_3})}{(p_1^{c_1}(1-p_1)^{1-c_1})(p_2^{c_2}(1-p_2)^{1-c_2})(p_3^{c_3}(1-p_3)^{1-c_3})} \end{aligned} \quad (7.4)$$

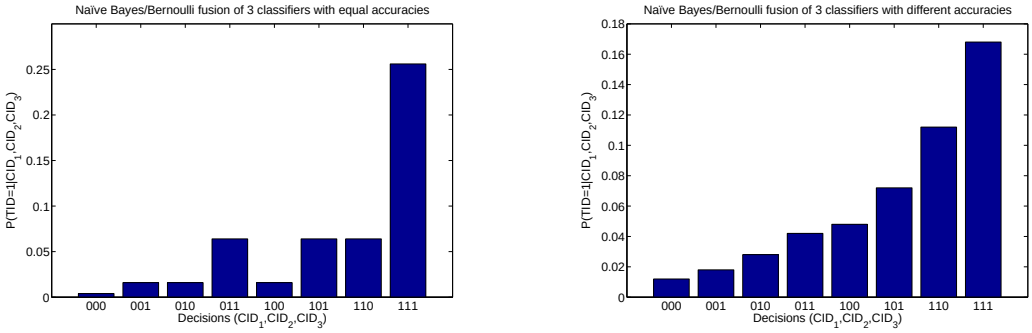
where the $p_l, l = 1 \ldots 3$ terms are obtained by marginalising over Ω in the $p_{l\omega}$ terms.

Note that if $c \neq t$ in $p_{l\omega}$, this quantity strictly corresponds to the false accept rate ($c = 1, \omega = 0$), respectively false reject rate ($c = 0, \omega = 1$) on the development set. As can be seen from Fig. 7.2(a), if all classifiers have the same accuracies ($\forall l$, $acc_l = acc_1$), then all decisions carry equal weight and all combinations of base classifier outputs in the set $\{(0,0,1),(0,1,0),(1,0,0)\}$ result in the same posterior probability. Indeed, since the "number of successes"* is 1 in all three cases, the standard binomial distribution is obeyed. If the base classifier accuracies are not equal, the posterior will look very different, and more accurate classifiers will influence the posterior more. This can be seen clearly from Fig.7.2(b), where the combination $(1,0,0)$ gives a higher posterior probability for $\Omega = 1$ than the combination $(0,0,1)$ since base classifier 1 has higher accuracy.

Score-level fusion

The probability distribution for continuous variables is typically represented as a normal distribution, and can be estimated in several ways, generally via a maximum likelihood estimates of the sufficient statistics of the distribution, but also using kernel methods [140], for example Parzen windowing [126]. Equation 7.28 shows the analytical form of the class-conditional density functions for

*in classical statistical theory terms; here this is equivalent to the number of base classifiers that have accepted the identity claim



(a) All base classifiers have the same accuracy ($acc_1 = acc_2 = acc_3$).

(b) The base classifiers have different accuracies. Here $acc_1 > acc_2 > acc_3$.

Figure 7.2 — Posterior probability $P(T|C_1, C_2, C_3)$ for naïve Bayes (Bernoulli product) decision-level fusion of 3 classifiers with equal (a) and different (b) classification accuracies acc_l. The prior is set to $P(\Omega = 1) = 0.5$.

a two-score fusion example, which is equivalent to using a multivariate Gaussian distribution with a diagonal covariance matrix.

In naïve Bayes modelling, training is simple (no hidden attributes), and inference is very efficient because of the tree structure of the network.

7.2.2 Tree-augmented naïve Bayesian network

The tree-augmented naïve Bayesian network (TAN) [93] seeks to take into account the existing correlation between features by using conditional mutual information, while still maintaining computational simplicity. To this end, "augmenting edges" can be added between features after the naïve Bayes model has been built, such that each feature node has at most one other feature and the class variable as parents. Thus, instead of having complete independence between features, we will have a dependence of the first order.

The procedure to build a TAN starts from a fully connected undirected graph where the nodes are the features, and the arcs between the nodes are weighted according to the conditional mutual information between the nodes linked by the arc given the class label. Then, arcs are dropped to form a maximum weighted spanning tree*. It is possible to set a threshold on mutual information so that some additional arcs are removed†. Then, a feature is chosen to be the root and the resulting graph is made directed, with arcs pointing from the root feature. Lastly, a class node is added along with edges pointing towards all features.

The joint probability for a TAN can be factored as:

$$P(\omega, X_1 = x_1, \ldots, X_l = x_l) = P(\omega) \prod_{l=1}^{L} P(X_l = x_l | \omega, pa(X_l)_{\setminus \omega}), \tag{7.5}$$

where the set of non-class parents $pa(X_l)_{\setminus \omega}$ has cardinality 1 or 0 and is trained by a structure learning algorithm.

*a spanning tree having maximum conditional mutual information between features. Several efficient algorithms exist to derive such trees (e.g.[303])

†See Section 7.4.5 for a possible approach.

Decision-level fusion

For decision-level fusion, the posterior consists of a scaled product of conditional binomial distributions. An example posterior for TAN fusion of 3 classifiers with $C_1 \rightarrow C_3$ and $C_1 \rightarrow C_2$ is

$$\begin{aligned} P(\omega|c_1,c_2,c_3) &= \frac{P(\Omega{=}\omega)P(C_1{=}c_1|\Omega{=}\omega)P(C_2{=}c_2|\Omega{=}\omega,C_1{=}c_1)P(C_3{=}c_3|\Omega{=}\omega,C_1{=}c_1)}{\sum_t P(\Omega{=}\omega)P(C_1{=}c_1|\Omega{=}\omega)P(C_2{=}c_2|\Omega{=}\omega,C_1{=}c_1)P(C_3{=}c_3|\Omega{=}\omega,C_1{=}c_1)} \\ &= \frac{(p_\omega^\omega(1{-}p_\omega)^{1-\omega})(p_{1\omega}^{c1}(1{-}p_{1\omega})^{1-c1})(p_{2\omega\gamma}^{c2}(1{-}p_{2\omega\gamma})^{1-c2})(p_{3\omega\gamma}^{c3}(1{-}p_{3\omega\gamma}^{1-c3})}{(p_1^{c1}(1{-}p_1)^{1-c1})(p_{2\gamma}^{c2}(1{-}p_{2\gamma})^{1-c2})(p_{3\gamma}^{c3}(1{-}p_{3\gamma})^{1-c3})}, \end{aligned} \tag{7.6}$$

where $p_{l\omega\gamma} \triangleq P(C_l = c|\Omega = \omega, pa(C_l)_{\setminus\Omega} = \gamma)$ is the probability that ensemble classifier l has taken value c given that the class is ω and the non-class parent has value γ. As for the naïve Bayes case of equation (7.4), the terms in the denominator are obtained by marginalising over Ω. Figure 7.3 presents an example of posterior for TAN decision-level fusion of three classifiers with equal accuracies.

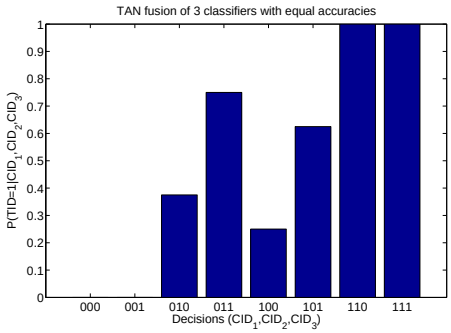


Figure 7.3 — Posterior probability $P(\Omega = 1|C_1, C_2, C_3)$ for TAN decision-level fusion of 3 classifiers with equal accuracies of 0.25. The prior is set to $P(\Omega = 1) = 0.5$.

As noted by Kuncheva [171], Section 4.6, the Chow-Liu algorithm [54] (a precursor of the TAN algorithm using mutual information instead of conditional mutual information) can be used to construct low-order approximations of discrete distributions for decision-level fusion. However, results reported in other fields of pattern recognition are generally not as good as with the TAN algorithm.

Score-level fusion

For score-level fusion, the TAN posterior consists of a scaled product of terms defined by Equation (4.1), where the conditional Gaussian density of each score will depend on the value of the parent score node.

An example TAN for score-level combination of 4 classifiers is shown in Fig. 7.1(b). In this model, it could be that classifier 1 (providing score variables Sc_1) is strongly correlated with classifiers 2 and 3 (providing Sc_2 and Sc_3), while classifier 2 is less correlated with classifier 3. Friedman et al. [93]'s original structure learning algorithm is designed for discrete data, but if the structure is to be learned automatically the score space can be discretised.

The TAN topology has been applied to combination problems by Davis et al. [61] in order to combine inductive logic programming rules, and found to clearly outperform naïve Bayes combination.

7.2.3 Other augmented variants of naïve Bayes

Many other variants on the naïve Bayes scheme can be used to perform multiple classifier fusion both at the score level and decision levels.

Langley and Sage [174]'s *selective Bayesian classifier* approach in fact refers to performing feature selection prior to building an NB classifier over the remaining features.

The general augmented naïve Bayesian network topology [93] starts from a naïve Bayes model and performs a search with a scoring criterion to find the best structure, with no limit on the result forming a tree over the feature nodes.

7.2.4 CART trees as Bayesian networks

Classification and regression trees (CART) [39] are a versatile family of classifiers, encompassing algorithms such as C4.5 that perform very well on a variety of tasks. Trees consists of query nodes and leaf nodes. A query node is a (typically binary) test on the value of a linear combination of features. A leaf node correponds to a class label.

To perform inference starting at the root node, at each query node, the result of the query decides which child node will be queried next. Once the current node is a leaf node, the class for the input pattern has been identified. The number of children per query node depends on the algorithm, but binary splits (two children per query node, yes/no answers) are sufficiently generic to handle all cases.

If the queries contain a single feature, the tree is called *monothetic*. In this case, segments of the decision boundaries will be perpendicular to the axes of the feature space. Given a tree that is sufficiently large, arbitrarily complex decision boundaries can be approximated [74].

Since monothetic trees with binary splits have sufficient representational power to handle complex pattern recognition tasks, we focus on establishing their representation as Bayesian networks.

7.2.5 Score-level fusion

An example of a CART tree is shown on Fig. 7.4 for score-level fusion of a fingerprint and a signature classifier on the BMEC 2007 development set.

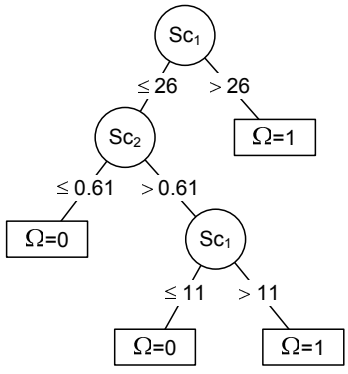


Figure 7.4 — Example CART for fusion of 2 classifiers. The data used in this example is taken from the BMEC 2007 development set, and Sc_1 corresponds to a fingerprint classifier score, while Sc_2 corresponds to a signature classifier score.

The simplest approach to implementing a Bayesian network version of CART trees for score fusion is to discretise the continuous input score space. Instead of having Q_l monothetic query

nodes on the score coming from classifier l, we divide the input score space from classifier l into R_l discrete bins. The bins need not be of equal size. The number of bins in which score Sc_l will be discretised is related to the number of query nodes on that variable in the equivalent CART tree by

$$R_l = Q_l + 1. \tag{7.7}$$

The binning defines a discrete joint probability space of dimension

$$D_L = \prod_{l=1}^{L} Q_l. \tag{7.8}$$

This can be represented as a conditional probability table (CPT) with $2D_L$ entries*, which is implemented as the Bayesian network shown in Fig. 7.5. As in other discrimative models, the joint probability is factored as

$$P(Sc_1, \ldots, Sc_L, \Omega) = P(Sc_1) \cdot \ldots \cdot P(Sc_L) P(\Omega | Sc_1, \ldots, Sc_L). \tag{7.9}$$

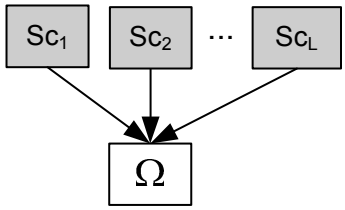


Figure 7.5 — Bayesian network topology for CART-like score-level fusion. Note the input scores have been discretised.

Assigning a class label to the entries in the CPT is performed in the standard manner by maximum likelihood. As with CART trees, it is possible to elicit a posterior probability rather than a crisp class label. The posterior probability of interest is trivially

$$\begin{aligned} P(\Omega | Sc_1, \ldots, Sc_L) &= \frac{P(Sc_1) \cdot \ldots \cdot P(Sc_L) P(\Omega | Sc_1, \ldots, Sc_L)}{\sum_\Omega P(Sc_1) \cdot \ldots \cdot P(Sc_L) P(\Omega | Sc_1, \ldots, Sc_L)} \\ &= \frac{P(Sc_1) \cdot \ldots \cdot P(Sc_L) P(\Omega | Sc_1, \ldots, Sc_L)}{P(Sc_1) \cdot \ldots \cdot P(Sc_L)} \\ &= P(\Omega | Sc_1, \ldots, Sc_L), \end{aligned} \tag{7.10}$$

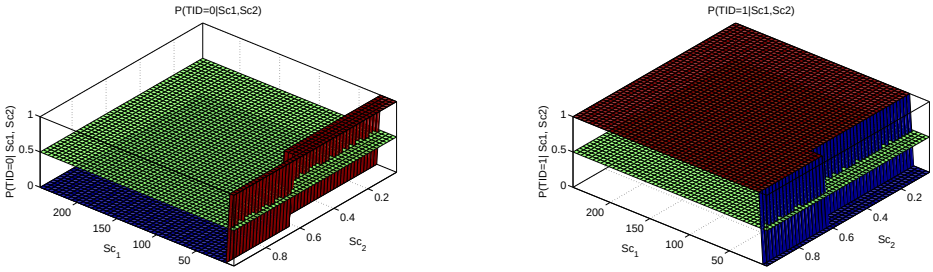
where the $2D_L$ probability terms are of the form $p_{l_{1\ldots L}\omega} \triangleq P(\Omega = \omega | Sc_1 = Sc_1, \ldots, Sc_L = Sc_L)$ and correspond to the parameters of a multinomial distribution.

The crisp (MAP-thresholded) posterior probability output $P(\Omega | Sc_1, Sc_2)$ of the BN classifier (Figure 7.5) corresponding to the example CART (Figure 7.4) is shown in Figure 7.6.

The bin boundaries corespond directly to the decision boudaries, as can be seen from the projection of Fig. 7.6 on the $S_1 - S_2$ plane.

A similar approach for continuous data is proposed by Woody and Brown [318], and Garg et al. [101] propose another architecture for a Bayesian network equivalent to a decision tree, albeit for discrete data. Ross and Jain [270] have used decision trees to fuse three biometric modalities.

*for each region of the score space we compte two probabilities, one for $P(\Omega = 1 | \mathbf{Sc})$ and one for $P(\Omega = 0 | \mathbf{Sc})$, trivially $1 - P(\Omega = 1 | \mathbf{Sc})$



(a) fused impostor posterior $P(\Omega = 0|Sc_1, Sc_2)$ (b) fused client posterior $P(\Omega = 1|Sc_1, Sc_2)$

Figure 7.6 — Monothetic CART density for two-classifier fusion. The left part shows the thresholded posterior probability for impostors, while the right part shows the thresholded posterior probability for clients.

7.3 Decision-level classifier combination with Bayesian networks

In this section, we cast several combination methods in probabilistic terms. This approach allows for precise comparison between decision-level combination methods, and show commonalities in many of these.

Using a probabilistic fomulation also allows for suggesting improvements to existing methods from a probabilistic perspective. An example is the setting of priors for the BKS combination scheme (see 7.3.2 to alleviate the curse of dimensionality inherent to this method.

Bayesian networks can be used to realise arbitrary boolean logic functions of binary variables. It is necessary to set their conditional probability tables to map the input variables to the output variables, assigning probabilities to the output O given the inputs I_n, thus resulting in a functional form $P(O|I_1, I_2, \ldots, I_N)$. In this case, the conditional probability table will result in the need to set 2^N values.

Thus, many decision combination rules found in the multiple classifier systems literature can be realised as a Bayesian network. We will present here state-of-the-art decision fusion schemes, namely majority vote, multinomial combination, error-correcting output coding, and decision templates (see Section 2.6). Indeed, for the problem of fusing multiple classifiers for a (two-class) biometric authentication, we will show some of these methods are strictly equivalent.

7.3.1 A Bayesian network for majority voting and Borda counts

Majority voting (equivalent to Borda counts on a 2-class problem [125]) can be seen as a logic function mapping input classifier opinion to an output decision according to the majority of classifiers. Taking as an example a 3-classifier fusion problem, the truth table corresponding to the majority vote function is shown in Table 7.1.

The majority vote function can be obtained by using the Bayesian network structure shown in Fig. 7.7.

The factorisation of the joint distribution for a three-classifiers version this graph is:

$$P(C_1, C_2, C_3, \Omega) = P(C_1)P(C_2)P(C_3)P(\Omega|C_1, C_2, C_3). \tag{7.11}$$

In this case the priors on the C_n nodes need not be specified (they can be set to uniform priors if required by the implementation), neither does the form of the probability distributions since these

variable	state of variable							
C_1	0	0	0	0	1	1	1	1
C_2	0	0	1	1	0	0	1	1
C_3	0	1	0	1	0	1	0	1
Majority vote	0	0	0	1	0	1	1	1

Table 7.1 — Truth table for Majority voting function. The C_n inputs correspond to the base classifier decisions, and the output correspond to the fused ensemble decision realising the majority vote function.

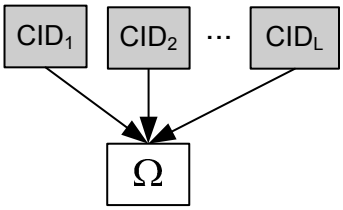


Figure 7.7 — Bayesian network model of majority voting with N classifiers.

random variables will always be observed and are modelled as exogenous variables. The posterior of interest is trivially given by

$$\begin{aligned} P(\Omega|C_1,C_2,C_3) &= \frac{P(C_1)P(C_2)P(C_3)P(\Omega|C_1,C_2,C_3)}{\sum_\Omega P(C_1)P(C_2)P(C_3)P(\Omega|C_1,C_2,C_3)} \\ &= \frac{P(C_1)P(C_2)P(C_3)P(\Omega|C_1,C_2,C_3)}{P(C_1)P(C_2)P(C_3)} \\ &= P(\Omega|C_1,C_2,C_3). \end{aligned} \tag{7.12}$$

Formally, the posterior corresponds to the multinomial probability of a single trial with 2^L possible outcomes, hence 8 in this example. It is not modelled as a product of binomials because we seek to model the dependency between "trials" (classifier outputs). The 2^L probability terms are of the form $p_{l_{1\ldots L}\omega} \triangleq P(\Omega=\omega|C_1=c_1,\ldots,C_L=c_l)$. Figure 7.8 shows the form of the posterior probability for this example.

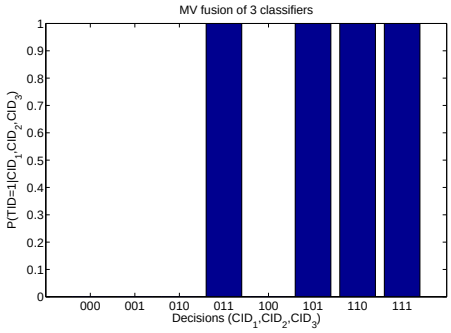


Figure 7.8 — Posterior probability $P(\Omega=1|C_1,C_2,C_3)$ for majority voting decision-level fusion of 3 classifiers.

The $P(\Omega|C_1,C_2,C_3)$ conditional probability table, corresponding to the $p_{l_{1,2,3}\omega}$ terms, is set as shown in Table 7.2. By comparing this specification of probabilities with the Majority vote function

specified in the truth table (Table 7.1), it can be seen that there is a direct correspondance between the two. Using this method, Bayesian networks can be used to realise any logic function of discrete variables.

C_1	0	0	0	0	1	1	1	1
C_2	0	0	1	1	0	0	1	1
C_3	0	1	0	1	0	1	0	1
$P(\Omega = 0 \mid C_1, C_2, C_3)$	1	1	1	0	1	0	0	0
$P(\Omega = 1 \mid C_1, C_2, C_3)$	0	0	0	1	0	1	1	1

Table 7.2 — Specification of the conditional probability table $P(\Omega|C_1, C_2, C_3)$ for majority vote using a Bayesian network.

7.3.2 Multinomial combination: a probabilistic implementation of the behaviour knoweldge-space method

By using a maximum-likelihood training procedure, the network shown in Fig. 7.7 can represent graphically an equivalent of the behaviour knowledge-space (BKS) method exposed in [132, 322]. In essence, BKS counts the number of times that a given classifier ensemble output values (say, in the above exemple, $(0, 1, 0)$) were assigned to a given class. Then, the class label most often seen in training is selected as the combined output. Thus, it can go beyond the majority vote because it is possible that a certain class is consistently misrecognised by the majority of classifiers. Likewise, it can learn the correlations between classifiers.

To implement BKS using Bayesian networks, it is necessary to learn the relevant multinomial parameters (conditional probabilities) by running a maximum likelihood learning algorithm (see Section 3.3.1) on a development set*. As per Equations (3.6) and (3.7), the sufficient statistics for the multinomial distribution are the occurence counts $\sum I_{l_{1\ldots L}\omega}$, where the indicator function $I_{\cdot}$ is one if the training vector correponds to base classifier decisions $C_1 = c_1, C_2 = c_2, \ldots, C_L = c_L$ with class value $\Omega = \omega$. Thus, the maximum likelihood estimate for each multinomial parameter is:

$$\hat{p}_{l_{1\ldots L}\omega} = \frac{\sum I_{l_{1\ldots L}\omega}}{\sum I_{l_{1\ldots L}}}. \tag{7.13}$$

The factorisation of the joint probability is the same as for majority voting (Eq.(7.11)), and the posterior of interest is still $P(\Omega|C_1, \ldots, C_L)$, only with probabilities learned on data instead of fixed a priori. Figures 7.9(a)-7.9(d) present various multinomial/BKS posteriors.

The class imbalance problem

Using Bayesian networks to represent a BKS combiner results in using a probabilistic approach based on relative frequencies (relative counts) rather than frequencies (counts) as in the case of the original BKS. This solves the important issue of class imbalance†: if one class has many more samples than another in the combiner training data, the counts of the ensemble deciding for that class will be higher than for other classes. This may be no problem if the testing set has the same structure, but in the case of biometric authentication the proportion of impostors to clients is unknown. Using

*In fact, the equivalence between BKS and this Bayesian network topology has been overlooked in the past, see for instance [101]. Raudys and Roli [242], Kuncheva [171], amongst others have noted the equivalence between the original formulation of BKS and the multinomial distribution.

†See also section 6.2.6 on the same issue in confidence measures for biometric authentication

relative counts solves the issue, since the counts are then normalised with respect to the number of training samples of each class.

The curse of dimensionality and parameter smoothing

One of the main weakness of the original BKS method is its intrinsic sensitivity to the curse of dimensionality. Indeed, some combinations of base classifier decisions may never occur in training. As the number of base classifiers increases, the problem is further compounded. After training some of the $p_{l_{1\ldots L}\omega}$ conditional probabilities may be left undecided. As an example, using the XM2VTS score database [233] containing 8 base classifiers, some 107 out of the 256 (2^8) possible decision combinations are unseen in training. Thus, we suggest that a reasonable probability estimate in this case is to use the posterior for majority voting*, with parameters found in Table 7.2. This will ensure that combinations of classifier outputs which do not occur in training are still modelled reasonably. Figure 7.9(a) shows an example on synthetic data for fusion of three classifiers where the problem of unseen combinations in training data appears. Figure 7.9(c) shows the same problem with a majority voting posterior substituted for the missing multinomial parameters.

Another problem of BKS is its tendency to overfit the training data [171]. A classic solution in taking a a probabilistic approach is to use parameter smoothing. Pseudo-counts are added to all combinations of base classifier decisions before $p_{l_{1\ldots L}\omega}$ is learned by maximum likelihood. That way, combinations of base classifier outputs that have rarely been seen in training can be given more importance. Figure 7.9(a) shows an example on synthetic data for fusion of three classifiers where overfitting is likely in the $(0, 0, 0)$ and $(1, 1, 1)$ cases. Figure 7.9(b) shows the posterior for the same fusion problem with Dirichlet priors added.

Using a Dirichlet prior, the maximum likelihood estimates of $p_{l_{1\ldots L}\omega}$ changes from Equation (7.13) to become:

$$\hat{p}_{l_{1\ldots L}\omega} = \frac{\alpha(l_{1\ldots L}, \omega) + \sum I_{l_{1\ldots L}\omega}}{\alpha(l_{1\ldots L}) + \sum I_{l_{1\ldots L}}}, \tag{7.14}$$

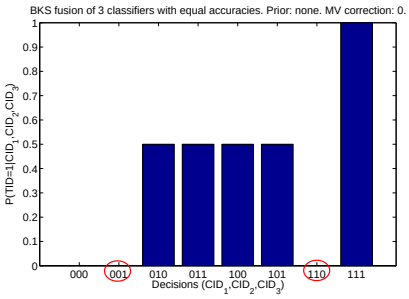
where the $\alpha(\cdot)$ parameters of a Dirichlet distribution can be regarded as prior counts.

Thus, by smoothing the distribution parameters and resorting to a majority vote posterior probability in case of lack of training data, the weaknesses of multinomial combination can be alleviated. At worse, if very sparse training data is provided, the multinomial combiner will still perform as the majority vote. As more training data is provided, the estimates are shifted away from their priors, and the maximum likelihood estimate of the smoothed multinomial parameters becomes asymptotically equivalent to the unsmoothed maximum likelihood estimates. In the limit of obtaining zero-bias estimates of the multinomial parameters ($p_{l_{1\ldots L}\omega}$ probabilities), the multinomial combination becomes the optimal combination rule [243].

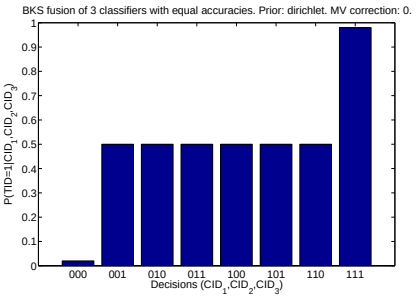
Other approaches for handling data sparsity

The augmented BKS (ABKS) combiner [55] deals with unseen base classifier output combinations by computing the confidence in each classifier's output, and setting the combined output to the most confident classifier's output. Unfortunately the method described for computing confidence does not allow for directly computing this value on classifiers with diverse output ranges such as MLPs and SVMs. A better way of obtaining good estimates of the competence of each classifier is to apply one of the methods described in Chapter 6 or to compute the reliability of each classifier's decision before allowing the most reliable to provide the final decision.

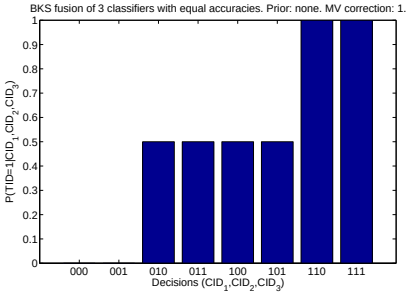
*An approach mentioned in [152], to which the probabilistic views of the BKS combiner and the majority vote function offer further support.



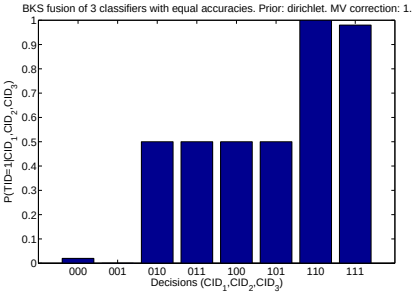
(a) posterior $P(\Omega = 1|CID_1, CID_2, CID_3)$ showing the training data sparsity problem: combinations $(0, 0, 1)$ and $(1, 1, 0)$ (circled on the horizontal axis) do not occur in the training set.



(b) Smoothed posterior $P(\Omega = 1|CID_1, CID_2, CID_3)$ with a Dirichlet prior. The effect of the prior on $(0, 0, 0)$ and $(1, 1, 1)$ has been exaggerated for visual clarity.



(c) posterior $P(\Omega = 1|CID_1, CID_2, CID_3)$ with majority voting posterior substituted for missing multinomial parameters



(d) posterior $P(\Omega = 1|CID_1, CID_2, CID_3)$ with Dirichler prior and majority voting posterior substituted for missing multinomial parameters. The effect of the prior on $(0, 0, 0)$ and $(1, 1, 1)$ has been exaggerated for visual clarity.

Figure 7.9 — Example multinomial fusion on a 3-classifiers synthetic data set. All base classifiers have 25% error rate, and some combinations do not occur in training.

Another, more complex approach to handling training data sparsity is to derive confidence intervals on the relative counts if the amount of training data is below a threshold, for instance less than $\frac{1}{10}$ of the amount of training samples of the least-represented class [313]*. Then, if one class has a lower confidence bound that is higher than all the other classes' upper confidence bound, that class is selected. If not, the best-performing (in terms of error rate on the training set) individual classifier in the ensemble is used to output the combined decision.

7.3.3 Error-correcting output coding based on a Bayesian network

Error-correcting output coding (ECOC) and Hamming distance computation can be used as a way of fusing decision-level classifier outputs [161]. This works by assigning a codeword to each class, the bits of which are the decisions of the L base learners. Thus, for the fusion of three classifiers, class 1 may be represented by codeword $(0, 1, 0)$ and class 2 may be represented by $(1, 0, 1)$. These codewords are chosen so as to maximise the Hamming distance (number of bits by which two codewords differ) between the codewords representing each class. Then, each base learner is trained using the corresponding codeword bit as target output (class output label). For instance, in the example above, classifier 1 would be trained to output 0 for class 1 and 1 for class 2. To perform fusion, the class whose codeword has the smallest Hamming distance to the vector of classifier output decisions is selected.

The maximal number of errors an error-correcting code with Hamming distance d can correct is [185, Ch. 13]

$$e_{max} = \lfloor (d-1)/2 \rfloor. \tag{7.15}$$

An interesting result for a two-class decison-level fusion problem is that the simple one-of-n encoding of classifier outputs used in majority voting† (see Section 7.3.1) constitutes an error-correcting code of maximal Hamming distance. For example, in a three-classifier fusion system, classifiers would all output 0 for class 1 and 1 for class 2. Thus, the codeword for class 1 would be $(0, 0, 0)$ and that for class 2 would be $(1, 1, 1)$. This has a Hamming distance of 3, meaning the system can cope with at most 1 error. Indeed, in the case of class 1 the majority vote for classifier outputs $(1, 0, 0)$, where one classifier is wrong, is still 0, but if two classifiers have taken erroneous decisions (say $(1, 0, 1)$), then the majority vote will also fail. In case of a tie, with an even number of classifiers, the majority vote will not be able to produce the correct output, as the "floor low" operator denotes in Eq. (7.15).

The majority voting Bayesian network presented in Fig. 7.7 therefore performs fusion equivalent to using an error-correcting output coding scheme, as the majority vote operation in this case is equivalent to selecting the class whose codeword minimises the Hamming distance to the vector of classifier decisions. Kuncheva [171], p.248 also notes the link between ECOC, majority voting, and decision templates.

The factorisation of the joint probability follows Eq. (7.11), the posterior is again a multinomial distribution with parameters $p_{l_{1,\cdots,L}\omega}$, and the posterior probability distribution looks like Figure 7.8.

In fact, for a two-class problem (such as biometric identity verification) the main difference between the majority voting approach and the ECOC approach is that the base classifiers in the ECOC scheme are trained on a different dichotomisation of the classes. For a two class-problem, this reduces to changing the labelling of the data for some of the base classifiers.

* The baseline form of this *coupling discriminator* method is the same as the BKS approach; the similarity is pointed out in [287]

† The codewords can however be chosen differently from a one-of-n encoding scheme, requiring only an adujstment in the specification of the conditional probability table $P(\Omega|C_1, \cdots, C_N)$.

7.3.4 Discriminative and generative models: Comparing naïve Bayes and voting-related schemes for fusion

Having formulated naïve Bayes, majority voting, BKS, ECOC, and decision template combiners as Bayesian networks, we can now look precisely at how these methods differ. For compactness in this section we will change variables and denote $C \stackrel{\triangle}{=} CID$.

Naïve Bayes fusion of three classifiers at the decision-level is expressed by the following decomposition of the joint probability density function:

$$P(T, C_1, C_2, C_3) = P(\Omega)P(C_1|\Omega)P(C_2|\Omega)P(C_3|\Omega). \tag{7.16}$$

The class posterior of interest (fused output of the ensemble) is

$$\begin{aligned} P(\Omega|C_1, C_2, C_3) &= \frac{P(\Omega, C_1, C_2, C_3)}{P(C_1, C_2, C_3)} \\ &= \frac{P(\Omega)P(C_1|\Omega)P(C_2|\Omega)P(C_3|\Omega)}{\sum_\Omega P(\Omega)P(C_1|\Omega)P(C_2|\Omega)P(C_3|\Omega)} \\ &= \frac{P(\Omega)P(C_1|\Omega)P(C_2|\Omega)P(C_3|\Omega)}{P(C_1)P(C_2)P(C_3)} \\ &= \alpha P(C_1|\Omega)P(C_2|\Omega)P(C_3|\Omega). \end{aligned} \tag{7.17}$$

In this case, the NB assumption that the base classifiers are independent means that during the learning phase we have $\forall(i,j),\ C_i \perp\!\!\!\perp C_j|\Omega$. Therefore, we only learn the feature-dependent marginals $P(C_i|\Omega)$ as a table which functions like a probabilistic confusion matrix.

The factored joint probability for majority-vote decision fusion with three classifiers is given in Eq. (7.11). In this case, the class posterior of interest is trivially

$$\begin{aligned} P(\Omega|C_1, C_2, C_3) &= \frac{P(\Omega, C_1, C_2, C_3)}{P(C_1, C_2, C_3)} \\ &= \frac{P(C_1)P(C_2)P(C_3)P(\Omega|C_1, C_2, C_3)}{\sum_\Omega P(C_1)P(C_2)P(C_3)P(\Omega|C_1, C_2, C_3)} \\ &= \frac{P(C_1)P(C_2)P(C_3)P(\Omega|C_1, C_2, C_3)}{P(C_1)P(C_2)P(C_3)} \\ &= P(\Omega|C_1, C_2, C_3). \end{aligned} \tag{7.18}$$

Thus, the combined output is directly given by conditional probability table associated with the class node Ω. In training, the dependencies between the classifiers C_i are learned, since as can be seen from the graph in Fig. 7.7, $\forall(i,j),\ C_i \not\perp\!\!\!\perp C_j|\Omega$.

Therefore, discrete models that have the class (Ω) node as parent to the feature nodes (CID) can be said to represent a generative modelling approach, while models that have the class node as a parent learn or define a decision boundary directly and thus can be said to represent a discriminative approach.

From the point of view of inference using a junction tree algorithm (Section 3.4.2), generative topologies offer more efficient inference than discriminative topologies. In the discriminative approach, clique sizes are larger because of the need to moralise and triangulate the graph since all decisions are parents of the class variable. This leads to larger potentials and less efficient inference.

7.4 Score-level classifier combination with Bayesian networks

In this section we show how Bayesian networks can be used for score-level fusion of multiple classifiers in the unimodal or multimodal context.

7.4.1 The product rule as a Bayesian network

Scores from different classifiers can be fused by the product rule, by which a product is taken over the scores and the resulting value is the fused score. This behaviour can be reproduced using Bayesian networks with the naïve Bayes topology shown in Fig. 7.1(a)[25]. Since the scores are assumed to be independent of each other ([151]), the factorisation is the same as in Equation(7.2). The form of the class-conditional score terms $P(Sc_l|\Omega)$ is assumed to be a univariate gaussian. The posterior probability for an example fusion of 2 classifiers is

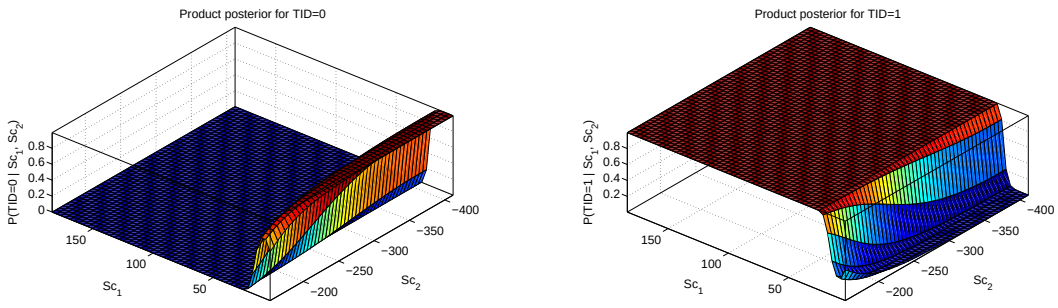
$$P(\Omega|Sc_1, Sc_2) = \frac{P(\Omega=\omega)\cdot\alpha_{\omega 1}e^{-\frac{1}{2}\left(\frac{Sc_1-\mu_{\omega 1}}{\sigma_{\omega 1}}\right)^2}\cdot\alpha_{\omega 2}e^{-\frac{1}{2}\left(\frac{Sc_2-\mu_{\omega 2}}{\sigma_{\omega 2}}\right)^2}}{\alpha_1 e^{-\frac{1}{2}\left(\frac{Sc_1-\mu_1}{\sigma_1}\right)^2}\cdot\alpha_2 e^{-\frac{1}{2}\left(\frac{Sc_2-\mu_2}{\sigma_2}\right)^2}} \tag{7.19}$$

$$= \frac{P(\Omega=\omega)\alpha_{\omega 1}\alpha_{\omega 2}}{\alpha_1\alpha_2}\cdot\frac{\left[(Sc_1-\mu_{\omega 1})^2\frac{1}{\sigma_{\omega 1}^2}+(Sc_2-\mu_{\omega 2})^2\frac{1}{\sigma_{\omega 2}^2}\right]}{\left[(Sc_1-\mu_1)^2\frac{1}{\sigma_1^2}+(Sc_2-\mu_2)^2\frac{1}{\sigma_2^2}\right]}, \tag{7.20}$$

where the variables indexed with ω are class-conditional, and the normalisation terms are

$$\alpha_{\omega l} \stackrel{\triangle}{=} \frac{1}{\sigma_{\omega l}\sqrt{2\pi}}. \tag{7.21}$$

Note that, as expected, this result is strictly equivalent to the use of a multivariate Gaussian density with diagonal covariance (See Equations (7.28) and Section 4.2). Figure 7.10 presents an example of posterior for the fusion of a fingerprint and a face classifier.



(a) fused impostor posterior $P(\Omega=0|Sc_1, Sc_2)$ (b) fused client posterior $P(\Omega=1|Sc_1, Sc_2)$

Figure 7.10 — Posterior probability $P(\Omega|Sc_1, Sc_2)$ for product rule fusion of two-classifiers (fingerprint and face) on BMEC 2007 data. The left part shows the posterior probability for impostors, while the right part shows the posterior probability for clients. Note that for the face classifier (Sc_2), less negative numbers indicate a better match

However, in the Bayesian network implementation, product combination cannot *stricto sensu* be considered as a fixed rule, since the posterior multiplies together *probabilities of scores*, not scores themselves. This offers the advantage of bounding the score terms between 0 and 1, which means the product will not be dominated by classifier outputs having large dynamic ranges. However, for

a practical implementation it implies a choice of a density function, Gaussian in the present case. If the distribution of scores is not strictly Gaussian, the (class-conditional and unconditional) mean and variance parameters may not be the appropriate sufficient statistics, and the fusion results may differ significantly from the simple implementation of the product rule as $Sc_1 \times Sc_2 \times \ldots \times Sc_L$.

7.4.2 Multivariate logistic regression

The score output from base classifiers can be considered as independent variables in a multivariate logistic regression (softmax) model. Instead of modelling joint distributions of variables $P(\Omega, Sc_1, Sc_2, \ldots, Sc_n)$ and extracting conditionals later on, we model the conditional probability $P(\Omega = \omega | Sc_1, Sc_2, \ldots, Sc_n)$ directly, taking a discriminative approach. The Bayesian network shown in Fig. 7.11 is used to perform logistic regression. This is similar to sigmoid belief networks [205, 292].

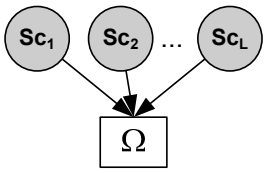


Figure 7.11 — Topology for score-level fusion with logistic regression

The probability density function of the Ω node is not implemented as a standard conditional probability table (which can accomodate only discrete data), but as a softmax density:

$$P(\Omega = \omega | Sc_1, Sc_2, \ldots, Sc_L) = \frac{e^{\mathbf{W}'_\omega \mathbf{Sc} + \mathbf{b}_\omega}}{\sum_\Omega e^{\mathbf{W}'_\omega \mathbf{Sc} + \mathbf{b}_\omega}}, \tag{7.22}$$

where the class-dependent weight matrix $\mathbf{W}_\omega$ and bias vector $\mathbf{b}_\omega$ can be trained efficiently using a Newton-Raphson algorithm called *iteratively reweighted least squares* [118, section 4.1.]. The use of softmax nodes allows for having continuous parents to discrete children.

Figure 7.12 shows the shapes of the densities learned using a Bayesian network for combining two classifiers at the score level via logistic regression fusion.

Logistic regression has been used by Pigeon et al. [228] for fusion of speaker verification classifiers.

7.4.3 Mixture of multivariate logistic regression functions

By using a mixture of logistic regressors [310], we can expect to obtain classification performance close to optimal [104], as each softmax density will be trained on a subset of training samples that form a cluster, and the mixing will then result in a better approximation of the "true" decision boundary; however this is achieved at the expense of increased complexity and training time. The softmax model of Figure 7.11 is augmented with a discrete hidden parent, as per Figure 7.13. The parameters of the hidden parent are then trained by using expectation-maximisation.

The joint probability can be factored as

$$P(\Omega, \mathbf{Sc}, M) = P(M)P(\mathbf{Sc})P(\Omega | \mathbf{Sc}, M), \tag{7.23}$$

where the $P(\Omega | \mathbf{Sc}, M)$ term is a softmax node as per Equation (7.22), only conditioned on the mixture variable M:

$$P(\Omega = \omega | Sc_1, Sc_2, \ldots, Sc_L, M) = \frac{e^{\mathbf{W}'_{m\omega} \mathbf{Sc} + \mathbf{b}_{m\omega}}}{\sum_\Omega e^{\mathbf{W}'_{m\omega} \mathbf{Sc} + \mathbf{b}_{m\omega}}}. \tag{7.24}$$

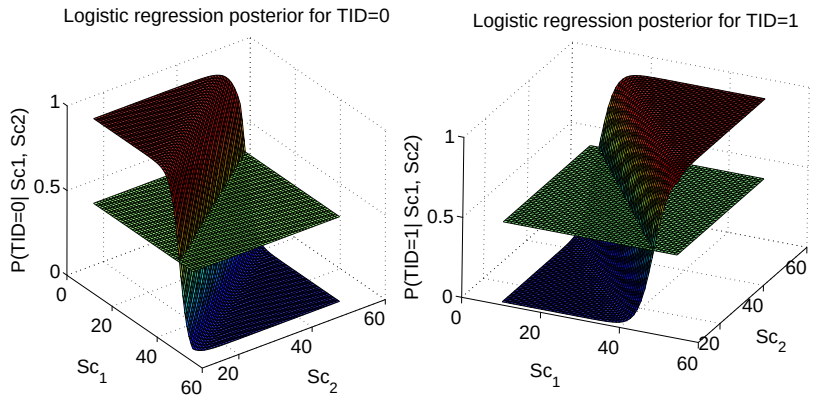


Figure 7.12 — Softmax density for two-classifier fusion. The left part shows the impostor posterior probability $P(\Omega = 0|Sc_1, Sc_2)$, while the right part shows the client posterior probability $P(\Omega = 1|Sc_1, Sc_2)$. The decision hyperplane is shown at 0.5.

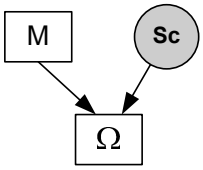
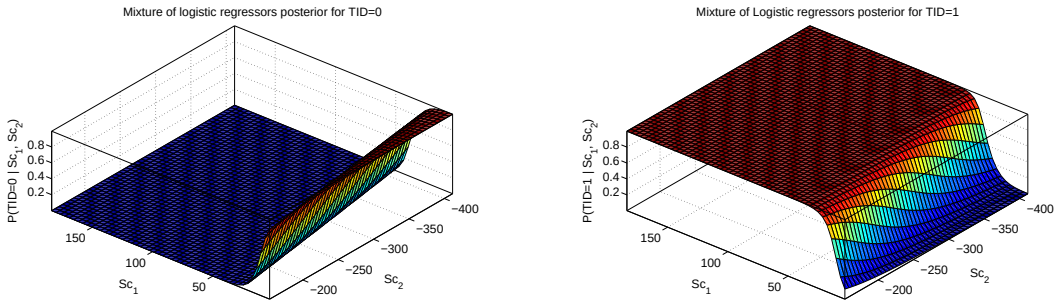


Figure 7.13 — Topology for score-level fusion using a mixture of logistic regressors. For compactness, the $Sc_1, \ldots, Sc_L$ base classifier outputs are represented as a single vector-valued score node.

The posterior probability is given by

$$P(\Omega|\mathbf{Sc}) = \frac{P(\Omega, \mathbf{Sc})}{P(\mathbf{Sc})} = \sum_M P(M)P(\Omega|M, \mathbf{Sc}). \tag{7.25}$$

The $P(M)$ term is initially set to a uniform distribution, and a weighted mixture can be obtained by learning the $P(M)$ distribution via expectation-maximisation. Thus, the posterior can be interpreted as a weighted average of the component softmax densities [160]. Figure 7.14 shows an example of mixture of logistic regressors fusion on two classifiers.



(a) fused impostor posterior $P(\Omega = 0|Sc_1, Sc_2)$ (b) fused client posterior $P(\Omega = 1|Sc_1, Sc_2)$

Figure 7.14 — Posterior probability $P(\Omega|Sc_1, Sc_2)$ for two-classifier fusion (fingerprint and face) on BMEC 2007 data using a mixture of two softmax densities. The left part shows the posterior probability for impostors, while the right part shows the posterior probability for clients. Note that for the face classifier (Sc_2), less negative numbers indicate a better match

7.4.4 Gaussian mixture model-based score fusion with Bayesian networks

Given the very good performance of Gaussian mixture models as base classifiers in biometric authentication, and their ability to approximate arbitraty probability density functions, It is expected that they should be a strong performer in classifier fusion as well.

The Bayesian network model equivalent to a Gaussian mixture model is shown in Fig. 4.3. The factorisation of the joint probability of the class label Ω, the hidden mixture weight variable M and the vector of scores to be fused $\mathbf{Sc}$ is

$$P(\Omega, M, \mathbf{Sc}) = P(\Omega)P(M|\Omega)P(\mathbf{Sc}|\Omega, M), \tag{7.26}$$

where the $P(M|\Omega)$ is a class-conditional multinomial distribution whose parameters are trained by expectation-maximisation.

An advantage of using Bayesian networks for performing probabilistic score-level fusion is that no normalisation of the scores from different classifiers is required prior to fusion. If each score is modelled as a separate node (as in the NB or TAN case, see Section 7.2), the variance and mean will be estimated on that score stream alone, and the output probability will in any case be bounded between 0 and 1. If the scores are modelled as vectors, a covariance matrix is estimated and its inverse is used to compute the distance between the model and the observed data (the Mahalanobis distance), thereby normalising each score individually by a value proportional to its variance. This property allows to avoid the reduction in performance for the combined system that can occur due to the loss of separability induced by normalisation [7].

The use of the Mahalanobis distance in the Gaussian mixture model is the key to explaining why GMM-based score fusion has very good performance compared to many other methods, if enough data is available to train the model parameters.

From Eq. (4.4), the probability density function for a Gaussian mixture component m used to fuse the components of a score vector $\mathbf{Sc}$ belong to class $\Omega = \omega$ is

$$p(\mathbf{Sc}|\Omega = \omega, M = m) = \underbrace{\frac{1}{|\mathbf{\Sigma}_{\omega m}|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}}}}_{\alpha_{\omega m}} e^{-\frac{1}{2}(\mathbf{Sc}-\boldsymbol{\mu}_{\omega m})'\mathbf{\Sigma}_{\omega m}^{-1}(\mathbf{Sc}-\boldsymbol{\mu}_{\omega m})}. \tag{7.27}$$

Assuming diagonal covariance and expanding the Mahalanobis distance for the case of fusion of scores from two classifiers yields

$$\begin{aligned} p(\mathbf{Sc}|\omega, m) &= \alpha e^{-\frac{1}{2}\left[\left(\begin{pmatrix} Sc_1 \\ Sc_2 \end{pmatrix} - \begin{pmatrix} \mu_{\omega m1} \\ \mu_{\omega m2} \end{pmatrix}\right)' \begin{pmatrix} \sigma^2_{\omega m1} & 0 \\ 0 & \sigma^2_{\omega m2} \end{pmatrix}^{-1} \left(\begin{pmatrix} Sc_1 \\ Sc_2 \end{pmatrix} - \begin{pmatrix} \mu_{\omega m1} \\ \mu_{\omega m2} \end{pmatrix}\right)\right]} \\ &= \alpha e^{-\frac{1}{2}\left[\begin{pmatrix} Sc_1 - \mu_{\omega m1} & Sc_2 - \mu_{\omega m2} \end{pmatrix} \begin{pmatrix} \frac{\sigma^2_{\omega m2}}{\sigma^2_{\omega m1}\sigma^2_{\omega m2}} & 0 \\ 0 & \frac{\sigma^2_{\omega m1}}{\sigma^2_{\omega m1}\sigma^2_{\omega m2}} \end{pmatrix} \begin{pmatrix} Sc_1 - \mu_{\omega m1} \\ Sc_2 - \mu_{\omega m2} \end{pmatrix}\right]} \\ &= \alpha e^{-\frac{1}{2}\left[(Sc_1-\mu_{\omega m1})^2 \frac{1}{\sigma^2_{\omega m1}} + (Sc_2-\mu_{\omega m2})^2 \frac{1}{\sigma^2_{\omega m2}}\right]}. \end{aligned} \tag{7.28}$$

Therefore, each mixture component (multivariate Gaussian) in a GMM performs weighted score fusion, where the importance of each score term is proportional to its distance from the mean and inversely proportional to its variance. Then, the multiplication of each mixture's output by the mixing coefficient c_m takes into account the amount of support of the mixture component, which is proportional to the responsibilities of this mixture component.

It should be noted that, if a single Gaussian component is used, forcing a diagonal covariance matrix in a multivariate Gaussian node is strictly equivalent to using a naïve Bayes model with Gaussian nodes for the continuous variables (as per Section 7.2.1). The use of several Gaussian components compensates for the independence assumption.

In the case of a full covariance matrix, off-diagonal terms are not zero and the following expansion is valid:

$$\begin{aligned} p(\mathbf{Sc}|\omega, m) &= \alpha e^{-\frac{1}{2}\left[\left(\begin{pmatrix} Sc_1 \\ Sc_2 \end{pmatrix} - \begin{pmatrix} \mu_{\omega m1} \\ \mu_{\omega m2} \end{pmatrix}\right)' \begin{pmatrix} \sigma^2_{\omega m1} & \sigma_{\omega m12} \\ \sigma_{\omega m12} & \sigma^2_{\omega m2} \end{pmatrix}^{-1} \left(\begin{pmatrix} Sc_1 \\ Sc_2 \end{pmatrix} - \begin{pmatrix} \mu_{\omega m1} \\ \mu_{\omega m2} \end{pmatrix}\right)\right]} \\ &= \alpha e^{-\frac{1}{2}\left[\begin{pmatrix} Sc_1 - \mu_{\omega m1} & Sc_2 - \mu_{\omega m2} \end{pmatrix} \begin{pmatrix} \frac{\sigma^2_{\omega m2}}{\sigma^2_{\omega m1}\sigma^2_{\omega m2}-\sigma^2_{\omega m12}} & -\frac{\sigma_{\omega m12}}{\sigma^2_{\omega m1}\sigma^2_{\omega m2}-\sigma^2_{\omega m12}} \\ -\frac{\sigma_{\omega m12}}{\sigma^2_{\omega m1}\sigma^2_{\omega m2}-\sigma^2_{\omega m12}} & \frac{\sigma^2_{\omega m1}}{\sigma^2_{\omega m1}\sigma^2_{\omega m2}-\sigma^2_{\omega m12}} \end{pmatrix} \begin{pmatrix} Sc_1 - \mu_{\omega m1} \\ Sc_2 - \mu_{\omega m2} \end{pmatrix}\right]} \\ &= \alpha e^{-\frac{1}{2}\left[(Sc_1-\mu_{\omega m1})^2 \frac{1}{\sigma^2_{\omega m1} - \frac{\sigma^2_{\omega m12}}{\sigma^2_{\omega m2}}} + (Sc_1-\mu_{\omega m1})(Sc_2-\mu_{\omega m2}) \frac{2}{\frac{\sigma^2_{\omega m1}\sigma^2_{\omega m2}}{\sigma_{\omega m12}} - \sigma_{\omega m12}} + (Sc_2-\mu_{\omega m2})^2 \frac{1}{\sigma^2_{\omega m2} - \frac{\sigma^2_{\omega m12}}{\sigma^2_{\omega m1}}}\right]}. \end{aligned} \tag{7.29}$$

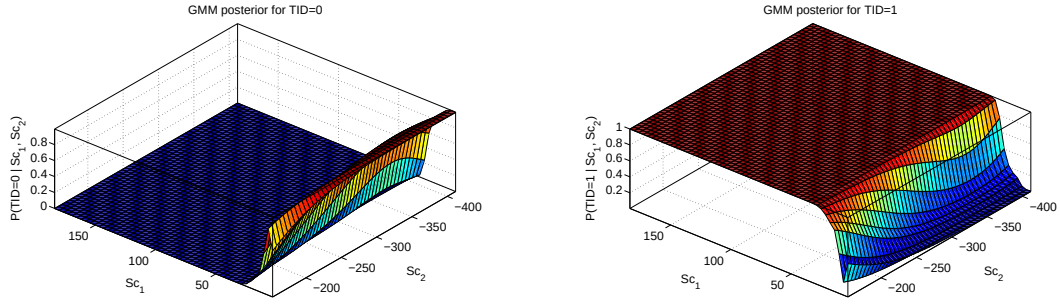
The GMM combiner for score-level fusion can be interpreted as a support-weighted sum of a variance-weighted sum of scores*. This interpretation offers a good intuition into why GMM-based fusion fails to outperform fixed rules when little data is available – the fusion weights are all dependent on variance, the estimate of which is biased if training data is too sparse.

*In a sense, each Gaussian mixture component is a classifier, and the mixture model output represents a consensus on the opinion of each individual mixture component.

The posterior probability of interest is:

$$\begin{aligned} P(\Omega|\mathbf{Sc}) &= \sum_M \frac{P(\Omega)P(M|\Omega)P(\mathbf{Sc}|\Omega,M)}{\sum_\Omega P(\Omega)P(M|\Omega)P(\mathbf{Sc}|\Omega,M)} \\ &= P(\Omega)\sum_M \frac{P(M|\Omega)P(\mathbf{Sc}|\Omega,M)}{P(M)P(\mathbf{Sc}|M)}. \end{aligned} \tag{7.30}$$

An example posterior for fusion of two scores with 4 diagonal covariance Gaussian components is shown in Fig. 7.15.



(a) fused impostor posterior $P(\Omega = 0|Sc_1, Sc_2)$ (b) fused client posterior $P(\Omega = 1|Sc_1, Sc_2)$

Figure 7.15 — Posterior probability $P(\Omega|Sc_1, Sc_2)$ for two-classifier fusion (fingerprint and face) on BMEC 2007 data using a a Gaussian mixture model with four diagonal-covariance Gaussian components. The left part shows the posterior probability for impostors, while the right part shows the posterior probability for clients. Note that for the face classifier (Sc_2), less negative numbers indicate a better match

7.4.5 Sparse regression score fusion with Bayesian networks

Instead of the mixture of (diagonal or otherwise) covariance matrices shown in section 7.4.4, which corresponds to the vector approach to multi-dimensional data modelling exposed in section 4.2, we can seek to learn conditional independence relationships in the score data. The principle of the sparse regression fusion algorithms (Algorithms 7.1 and 7.2) is to model only relationships between variables which are dependent. The level of model complexity can be set either via a threshold on the dependence criterion, or via a pre-specified number of edges. Because of the equivalence between a covariance representation of multivariate data and the regression approach we take here (see Section 4.2.3), the basic principle of the algorithm can be seen as equivalent to the procedure of Dempster [66] which increases model sparsity by forcing zeros in the correlation or inverse covariance matrix.

Figure 7.16 shows an example of a fully-connected DAG for combination of 4 classifiers. In this case, all continuous variables being observed, no indepence relationship between the classifier outputs is deemed to exist. This fully connected model corresponds to a full covariance matrix in multivariate modelling.

This model has a large number of parameters: for each edge coming from a continuous parent into a continuous node, we need to learn an additional regression weight. A fully connected DAG has a number of edges equal to

$$|E_{full}| = \frac{L^2}{2} - \frac{L}{2}, \tag{7.31}$$

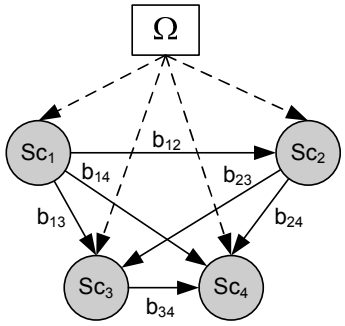


Figure 7.16 — Fully connected (full regression) fusion model for 4-classifier combination

where L is the number of continuous variables (scores) to be fused. In addition, the connection from the class node to the scores (naïve Bayes structure) adds

$$|E_{NB}| = L. \tag{7.32}$$

edges.

Assuming a fixed sample size T, the number of parameters to be learned increases according to Equation (7.31) with the number of classifiers in the ensemble. Training a large amount of parameters over a small amount of data may lead to learning some spurious dependence, which will hinder the generalisation performance of the fusion model. We wish to model only the most important relationships in the data set.

Vanhoucke and Sankar [308] have studied the effect of removing elements from covariance matrices modelling audio features and found that nearly the same classification accuracy could be obtained while keeping about half the number of elements. This is in line with the findings reported by Bilmes [24], where keeping about 30% of parameters results in virtually identical error rates. We go further, and show in the experimental results of Section 7.5.2 that it is possible to obtain *better* performance in terms of error rates than with a more complex model by not jointly modelling independent classifiers.

Measuring dependencies between classifier outputs

Thus, we need a measure of dependency between classifier score-level outputs. One simple measure, the Pearson linear correlation coefficient, is not appropriate for the task for two main reasons already evoked in Section 5.4 with respect to quality measures:

- The relationship between random variables is assumed to be linear. Figure 7.17 shows an typical example where this assumption does not hold.
- The random variables are supposed to be have homoscedastic distributions. Again, this is rarely the case in practice with scores coming from biometric verification classifiers. This is also exemplified in Figure 7.17.

Measuring conditional dependencies between classifier outputs

In order to properly assess independence in Bayesian networks, it is necessary to be able to measure *conditional* independence.

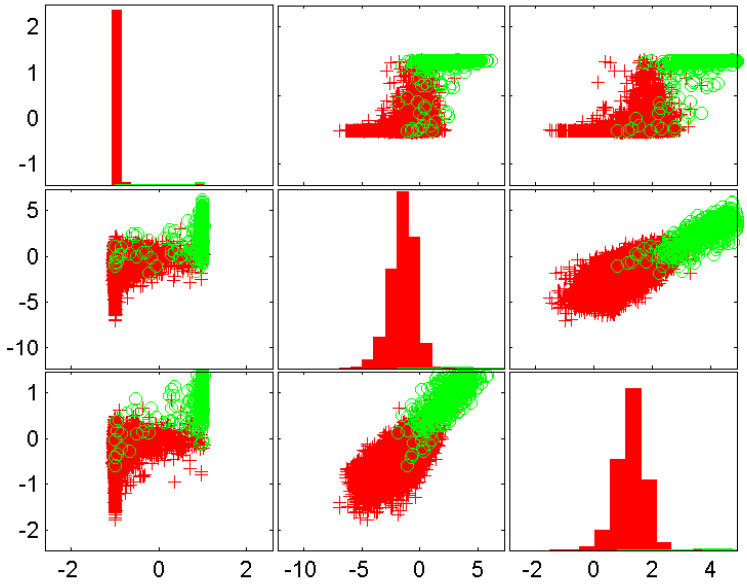


Figure 7.17 — Scatterplot for the scores of 3 face classifier on XM2VTS (data from [233]). Green circles indicate client accesses, and red crosses indicate impostor accesses.

While Spirtes et al. [293], p.47 propose using partial correlation coefficients to assess the *conditional* independence between random variables, Baba et al. [13] argue that "conditional independence has no close ties with zero partial correlation except in the case of the multivariate normal distribution; [...]". Therefore, a second measure of independence we use is conditional mutual information.

Having a measure of conditional mutual information is especially important in biometric applications, where the (class-unconditional) score distributions for clients and impostor are typically strongly heteroscedastic, as can be seen in Figure 7.17.

Base classifier independence and graph topology: application of the d-separation rules

As explained in Section 3.2.5, if a distribution is faithful to a directed acyclic graph $\mathcal{G}$, stating that $X \perp\!\!\!\perp Y|Z$ (as measured by $\bar{I}(X;Y|Z)$) implies that $\langle X|Z|Y\rangle_{\mathcal{G}}$ (X is d-separated from Y in $\mathcal{G}$ given Z). In the sparse regression fusion model, all scores constitute observed variables, and all are children of the class node, which is also observed in training. Thus, if $Sc_i \perp\!\!\!\perp Sc_j|\Omega$, we want the corresponding Bayesian network topology $\mathcal{G}$ to have $\langle Sc_i|\Omega|Sc_j\rangle_{\mathcal{G}}$ and consequently no edge $Sc_i \rightarrow Sc_j$ or $Sc_i \rightarrow Sc_j$ is allowed to exist in $\mathcal{G}$.

However, perhaps less intuitively, if Sc_i and Sc_j share a common child Sc_c (we have $Sc_i \rightarrow Sc_c$ and $Sc_j \rightarrow Sc_c$), we have $Sc_i \not\perp\!\!\!\perp Sc_j|Sc_c$. Therefore, it is not sufficient to remove direct edges between variables that are deemed to be independent, but care must be taken if the two variables have children. Otherwise, the child node distribution will be a function (regression) of the two parent node distributions.

Figure. 7.18 shows an example where the conditional independence relationships $Sc_1 \perp\!\!\!\perp Sc_2|\Omega$ and $Sc_2 \perp\!\!\!\perp Sc_3|\Omega$ are encoded in the sparse regression graph. Note that in this case, Sc_4 is a common child of Sc_1 and Sc_2, and thus we have $Sc_1 \not\perp\!\!\!\perp Sc_2|Sc_4$. If we want to further enforce $Sc_1 \perp\!\!\!\perp Sc_2|\{Sc_4, \Omega\}$, either the $Sc_1 \rightarrow Sc_4$ or the $Sc_2 \rightarrow Sc_4$ arc needs to be removed. The final

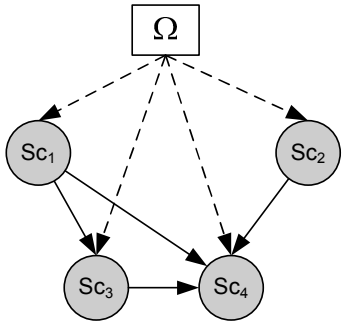


Figure 7.18 — Example (intermediate) sparse regression fusion model for 4-classifier combination

model is shown in Figure 7.19.

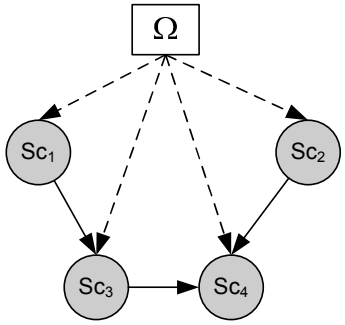


Figure 7.19 — Example final sparse regression fusion model for 4-classifier combination

Modelling non-normal score data

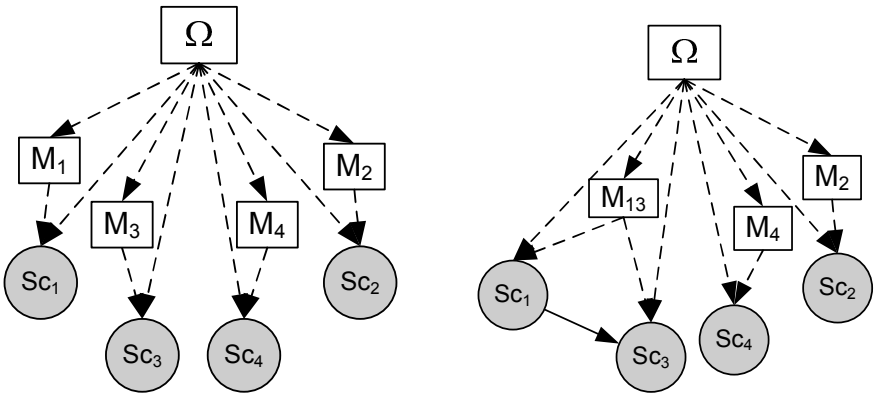
It is often the case that the distribution of score outputs from a base classifier deviates substantially from a Gaussian assumption.

Thus, as explained for the mixture model of 4.3, we add a discrete hidden parent to each score node in the model, the cardinality of which corresponds to the number of Gaussian components in the mixture. If no arcs are present between score nodes (the classifiers are deemed independent), this initial topology is equivalent to a product of univariate mixtures of Gaussian densities – equivalent to the naïve Bayes network of Figure 7.2.1, only with mixture densities. This is shown on Figure 7.20(a).

However, if a subset of score nodes is connected either in a chain configuration (head-to-tail), or in a collider configuration (v-shape), we supress the individual mixture nodes and create a single mixture node as a a common parent of the classifiers in the subset (see the example on Figure 7.20(b) for a simple chain). In learning, this has the effect of causing the EM algorithm to compute the expected sufficient statistics and to maximise the likelihood with respect to the score nodes in the subset *jointly*, rather than separately for each score node.

The sparse regression fusion algorithm

To summarise, if two classifiers outputs are independent given the class, as indicated by low conditional mutual information, there is little to be gained by modelling their joint density. In this case, no arc should be included between the two. However, a product-style combination is motivated by probability theory. Thus, in the sparse regression fusion model, the factorisation of the joint density



(a) Independence between classifiers leads to Naïve Bayes fusion with mixture densities.

(b) Classifiers 1 and 3 are not independent, they share a hidden parent.

Figure 7.20 — Two examples of sparse regression fusion model with mixture score modelling.

of all classifiers is a product of first-order dependencies (only the class is in the conditioning set) and of higher-order dependencies, for classifiers that are correlated.

The method exposed above can be implemented either as a forward or a backward procedure, as formalised in Algorithms 7.1 and 7.2.

Algorithm 7.1 Sparse Regression Fusion algorithm (SRF) for continuous data: forward version

1: Drafting: start with a naïve Bayes DAG $\mathcal{G} = (V, E)$
2: Criterion computation: for all pairs of variables (Sc_i, Sc_j) where $j \neq i, j > i$, compute $\bar{I}(Sc_i; Sc_j|\Omega)$.
3: Thickening: Add $Sc_i \rightarrow Sc_j$ regression edges to the E where $\bar{I}(Sc_i; Sc_j|\Omega) > \bar{I}_\tau$.
4: Criterion computation 2: for all pairs of variables (Sc_i, Sc_j) where $j \neq i$, if $\exists Sc_k, E_{Sc_i Sc_k} \in E, E_{Sc_j Sc_k} \in E$, compute $\bar{I}(Sc_i; Sc_j|\Omega, Sc_k)$.
5: Thinning: for score pairs that have $\bar{I}(Sc_i; Sc_j|\Omega, Sc_k) < \bar{I}_\tau$, compute $\bar{I}(Sc_i; Sc_k|\Omega)$ and $\bar{I}(Sc_i; Sc_k|\Omega)$, and remove from E the corresponding edge $Sc_{i,j} \rightarrow Sc_k$ that has the lowest value.
6: Mixture nodes addition: Find all maximal length undirected paths $P_{ik} = Sc_i, \ldots, Sc_k$, over nodes connected by regression edges. For each path, add a mixture node as a common parent to all nodes in the path.

Once the structure of the model is trained, the joint probability can be factored as follows:

$$P(\Omega, Sc_1, \ldots, Sc_l) = P(\Omega) \prod_{l=1}^{L} P(Sc_l|\Omega, pa(Sc_l)_{\setminus\Omega}), \tag{7.33}$$

where $pa(Sc_l)_{\setminus\Omega}$ is the set of non-class parents of the score nodes, which for all score nodes has a cardinality $0 \leq |pa(Sc_l)_{\setminus\Omega}| \leq L - 1$.

Choosing a numerical threshold for independence

In theory, only classifier outputs with zero (conditional) mutual information should be considered independent. However, given that the size of the training sample is limited and we may make wrong modelling assumptions in computing the joint densities, in practice even the realisations of

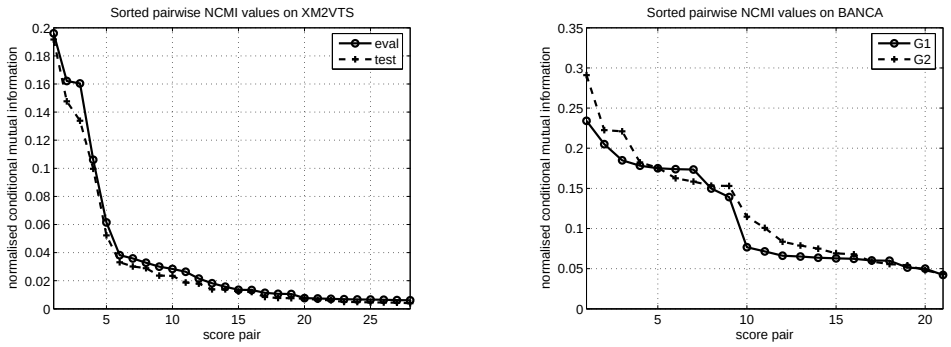
Algorithm 7.2 Sparse Regression Fusion algorithm (SRF) for continuous data: backward version

1: Drafting: start with a fully connected DAG $\mathcal{G} = (V, E)$
2: Criterion computation: for all pairs of variables (Sc_i, Sc_j) where $j \neq i, j > i$, compute $\bar{I}(Sc_i, Sc_j|\Omega)$.
3: Thinning: Remove all edges from DAG where $\bar{I}(Sc_i; Sc_j|\Omega) < \bar{I}_\tau$.
4: Criterion computation 2: for all pairs of variables (Sc_i, Sc_j) where $j \neq i$, if $\exists Sc_k, E_{Sc_i Sc_k} \in E, E_{Sc_j Sc_k} \in E$, compute $\bar{I}(Sc_i; Sc_j|\Omega, Sc_k)$.
5: Thinning: for score pairs that have $\bar{I}(Sc_i; Sc_j|\Omega, Sc_k) < \bar{I}_\tau$, compute $\bar{I}(Sc_i; Sc_k|\Omega)$ and $\bar{I}(Sc_i; Sc_k|\Omega)$, and remove from E the corresponding edge $Sc_{i,j} \rightarrow Sc_k$ that has the lowest value.
6: Mixture nodes addition: Find all maximal length undirected paths $P_{ik} = Sc_i, \ldots, Sc_k$, over nodes connected by regression edges. For each path, add a mixture node as a common parent to all nodes in the path.

independent random variables may give rise to non-null (conditional) mutual information. Thus, it becomes necessary to set a numerical threshold under which two variables are to be considered independent.

We propose to set the independence threshold $\bar{I}_\tau$ by computing the *normalised conditional mutual information map*, a matrix of $\bar{I}(Sc_i; Sc_j|\Omega)$ values. Then, by sorting the values in decreasing order, it is possible to understand where the cutoff point should be set, by trying to achieve a trade-off between number of arcs and total conditional mutual information preserved.

Fig. 7.21 shows such plots for the XM2VTS and BANCA databases. For XM2VTS, it suggests that the optimal number of arcs should be between 5 and 7 (containing most of the mutual information, corresponding to a threshold of about 0.04, or about 10% of the arcs in the fully connected model, in turn suggesting overall low dependence between classifiers. For BANCA, keeping 8 to 10 arcs (corresponding to a threshold between about 0.15 to 0.07) would conserve most of the "mutual information mass". This is about 30% of the arcs in the fully connected model, suggesting higher dependence between classifiers.



(a) Normalised conditional mutual information values for XM2VTS

(b) Normalised conditional mutual information values for BANCA

Figure 7.21 — Exhaustive set of values of $\bar{I}(Sc_i; Sc_j|\Omega)$ for all base classifier score pairs in the ensemble, sorted in decreasing order. Note that the cliff effect appears at a different number of pairs depending on the classifier ensemble and database. Also note that the vertical scale of the graphs is different.

Classifier dependencies in unimodal and multimodal multi-classifier systems

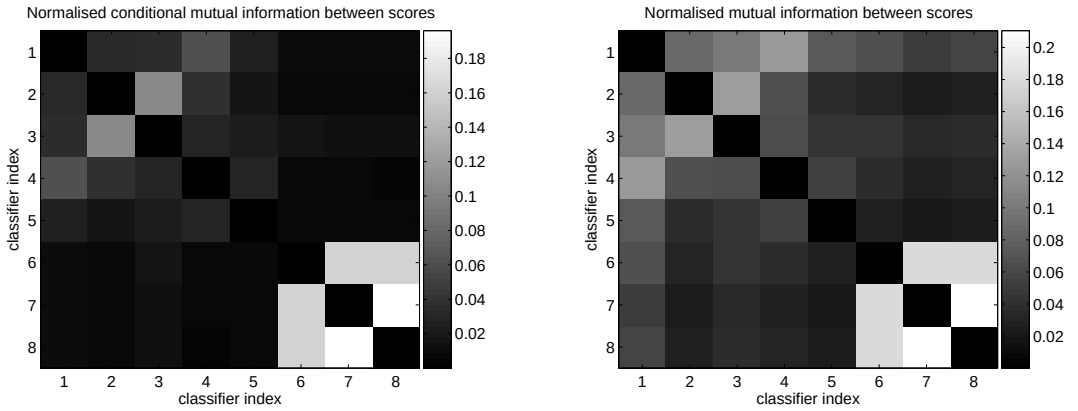
It is not true that classifier outputs are *independent* if they are trained on signals from different modalities: any well-behaved biometric classifier will have a (conventionally) higher output score if the person the data belongs to is a client than if it is an impostor, whatever the modality. However, it can be argued that classifier outputs should be *conditionally* independent (given the class) if they are from different modalities.

We expect to find a reflection of this principle in measures of mutual information and conditional mutual information on real data: The average normalised mutual information between (reasonably trained) classifiers of different modalities $\bar{I}(Sc_i; Sc_j)_{\mu b}$ should be proportionately higher than the average normalised conditional mutual information between the same classifiers given the class $\bar{I}(Sc_i; Sc_j|\Omega)_{\mu b}$.

An exemple on real data is shown on Fig. 7.22 for 5 face and 3 speech classifiers running on the XM2VTS database. The figure shows higher average normalised mutual information and normalised conditional mutual information within-modality than between-modality, as well as a proportionately higher difference between within-modality and between-modality classifiers for normalised conditional mutual information. Table 7.3 summarises the numerical results on BANCA and XM2VTS.

database	$\bar{I}(\cdot;\cdot)_{\mu 1}$	$\bar{I}(\cdot;\cdot)_{\mu 2}$	$\bar{I}(\cdot;\cdot)_{\mu b}$	$\bar{I}(\cdot;\cdot)_{\mu w}$	$\bar{I}(\cdot;\cdot)_{\mu b}/\bar{I}(\cdot;\cdot)_{\mu w}$
XM2VTS (eval)	0.079	0.189	0.036	0.104	0.348
BANCA (G1)	0.261	0.359	0.163	0.294	0.554
	$\bar{I}(\cdot;\cdot\|\Omega)_{\mu 1}$	$\bar{I}(\cdot;\cdot\|\Omega)_{\mu 2}$	$\bar{I}(\cdot;\cdot\|\Omega)_{\mu b}$	$\bar{I}(\cdot;\cdot\|\Omega)_{\mu w}$	$\bar{I}(\cdot;\cdot\|\Omega)_{\mu b}/\bar{I}(\cdot;\cdot\|\Omega)_{\mu w}$
XM2VTS (eval)	0.040	0.173	0.009	0.071	0.129
BANCA (G1)	0.182	0.174	0.061	0.179	0.340

Table 7.3 — Average normalised mutual information $\bar{I}(Sc_i; Sc_j)_{\mu\cdot}$ and average normalised conditional mutual information $\bar{I}(Sc_i; Sc_j|\Omega)_{\mu\cdot}$ for modality 1 (face, subscripted $_{\mu 1}$), modality 2 (speech, $_{\mu 2}$), between-modality ($_{\mu b}$), within-modality ($_{\mu w}$), and ratio of between-to-within-modality (last column)



(a) Normalised conditional mutual information map showing $\bar{I}(Sc_i; Sc_j|\Omega)$ between all pairs of classifiers.

(b) Normalised mutual information map showing $\bar{I}(Sc_i; Sc_j)$ between all pairs of classifiers.

Figure 7.22 — Normalised conditional mutual information and normalised mutual information maps for the score outputs from 5 face classifiers (indices 1-5) and 3 speech classifiers (indices 6-8) on XM2VTS Lausanne Protocol 1 [233]. Note that for display purposes the computation of the value for the classifier with itself ($i = j$ case) has been set to 0, rather than its normal value of 1.

This observation is an explanation of typical outputs of the SRF algorithm on multi-classifier multimodal data: the arcs between classifiers for the same modality generally carry a higher conditional mutual information weight than the arcs between classifiers for different modalities.

7.4.6 Discriminative and generative models in score-level fusion

Again, fusion models for score-level fusion can be divided into two broad types depending on whether the class node has parents or not. The product rule/naïve Bayes, TAN model, Gaussian mixture model and Sparse regression fusion models all have the class node as parent to score nodes, and thus represent a generative approach to classification.

The CART tree, multivariate logistic and mixture of multivariate logistic models seek to directly estimate the decision boundaries, and represent a discriminative approach, as hinted by the functional form of the Ω node conditional probability, $P(\Omega|Sc_1, \ldots, Sc_L)$.

7.5 Experiments and results

In this section we present results of fusion experiments using three multimodal databases of scores.

The first database is the Poh and Bengio [233] database of scores on XM2VTS, consisting of 2 modalities, speech and face, with respectively 5 and 3 classifiers. This database contains 295 users, for a total of 40'600 score vectors in the eval set and 112'200 score vectors in the test set. The protocol used is the Lausanne protocol, configuration 1. The results are reported by training the fusion models on the evaluation set and testing them on the testing set.

The second database is the BMEC 2007 fusion development database experiment 2, consisting of 3 modalities, signature (HMM model, Biosecure reference system), face (eigenfaces approach, Biosecure reference system), and fingerprint (minutiae-based, NIST system), each with one classifier. This database contains 50 users, each with 4 client and 20 impostor accesses, for a total of 1200 score vectors. Given the absence of a mandatory evaluation protocol on this database, the results are reported by performing 4-fold cross validation.

The third database is the IDIAP repository of scores for the BANCA database [14], originally consisting of 2 speech classifiers and 3 face classifiers, but augmented with one additional speech classifier and one additional face classifier[167]. This database contains 2x26 users, for a total of 546 score vectors in G1 and 546 score vectors in G2. The protocole followed is the P protocol. The results are reported by taking an average of measures when first training the fusion model on G1 and testing on G2, then training on G2 and testing on G1.

7.5.1 Decision-level fusion

Experimental setup

For decision-level fusion, we train the decision thresholds a priori on each training dataset, and they are set to the EER threshold.

The TAN model structure is learned using normalised conditional mutual information (see Section 7.4.5) instead of the standard conditional mutual information as per the original Friedman et al. [93] algorithm.

The classifiers based on Bayesian networks are compared for reference with a multi-layer perceptron classifier and a kernel-based classifier. The kernel-based classifier is a support vector machine with polynomial kernels, trained using sequential minimal optimisation [231]. We use the Weka implementation [314] for both.

Results and discussion on XM2VTS

Results for XM2VTS are shown in Table 7.4.

The good performance of majority voting can be attributed to the generally low correlation between classifiers, as exemplified in Figure 7.21.

Overall, it can be seen that Bayesian network-based classifiers are competitive with state-of-the-art classifiers such as MLPs and SVMs.

The worse results for the majority voting correction (see Section 7.3.2) applied to the multinomial combination than for the multinomial combination can be explained by the fact that, in absence of data, the posterior probability of any class will be 0.5 (equal to the Dirichlet prior). Since the decision threshold is set at *greater than* 0.5, all unseen combinations will be treated as impostor accesses. Since impostor accesses are indeed the majority class in the training set (about a 280:1 ratio), it is more likely that labelling an access as an impostor access is correct.

In general, large discrepancies between EER and HTER can be attributed to the cardinality of the output of the fusion model, as some have only a few possible output values, while others have a continuous range. This potential scarcity of fusion output values entails having few points on the DET curve, which in turn means the EER point will be badly computed.

fusion classifier	**err** [%]	**FAR** [%]	**FRR** [%]	**HTER** [%]	**EER** [%]
base best (face)	1.04	1.04	1.25	1.14	1.14
BN/Majority Voting	0.09	0.086	1.25	0.67	0.67
BN/Bernoulli	0.082	0.078	1.25	0.66	0.45
BN/TAN	0.026	0.020	1.75	0.88	0.37
BN/multinomial	0.022	0.009	3.75	1.88	0.38
BN/multinomial (MV corr)	0.053	0.044	2.50	1.27	0.58
MLP	0.032	0.022	2.75	1.39	0.42
SVM	0.025	0.017	2.25	1.13	1.13

Table 7.4 — Results of decision-level fusion models on the XM2VTS database.

Results and discussion on BMEC 2007

Another set of experiments is performed on the BMEC 2007 development database of scores, which contains 1 classifier per modality in 3 modalities. The somewhat surprising results shown in Table 7.5 (Bernoulli, TAN, and multinomial combination give the same results) can be attributed to the fact that there is a nearly inexistent dependence between the classifiers decisions once the class is known (The signal comes from 3 distinct and *a priori* unrelated modalities: signature, speech, and fingerprint). On this database, the largest normalised conditional mutual information between decisions is vanishingly small at 0.005.

For the TAN model, over the 4 folds of the cross-validation, the maximum spanning tree weight (sum of normalised conditional mutual informations) obtained by the TAN model is 0.01. Because the classifier decisions are for all practical purposes independent, the conditional binomial distributions reduce to Bernoulli distributions since the conditioning terms (other classifier decisions) do not affect the probability estimates.

The majority voting posterior correction applied to the multinomial combiner (Section 7.3.2) has no effect here because there are only 3 classifiers in the ensemble and no combination of decisions in unseen in training data.

fusion classifier	err [%]	FAR [%]	FRR [%]	HTER [%]	EER [%]
base best (fingerprint)	13.08	13.10	13.00	13.05	13.05
BN/Majority Voting	8.42	7.80	11.50	9.65	9.65
BN/Bernoulli	5.33	1.20	26.00	13.60	10.35
BN/TAN	5.33	1.20	26.00	13.60	10.35
BN/multinomial	5.33	1.20	26.00	13.60	10.35
BN/multinomial (MV corr)	5.33	1.20	26.00	13.60	10.35
MLP	5.33	1.20	26.00	13.60	10.35
SVM	5.33	1.20	26.00	13.60	13.60

Table 7.5 — Results of decision-level fusion models on the BMEC 2007 database. Note the EER result for SVM is a computation artefact due to the small cardinality of the possible output values.

Results and discussion on BANCA

The results for BANCA, shown in table Table 7.6, again exemplify the competitive performance of the Bayesian network-based fusion algorithms, which perform very close to state-of-the art methods, with the Bernoulli combiner bringing the best results.

The failings of the multinomial combiner is evident, as the size of the training data for BANCA is not sufficient with respect to the dimensionality of the space (7 classifiers). In this case, the majority voting correction help overcome the sparsity of the data, and succeed in bringing the error rate well below that of the best baseline classifier and the majority voting combiner.

The Bernoulli combiner is a very strong performer, as it does not suffer as much as multivariate methods from the curse-of-dimensionality due to the lack of training data in BANCA with respect to the number of classifier to fuse.

fusion classifier	err [%]	FAR [%]	FRR [%]	HTER [%]	EER [%]
base best (speech)	4.49	4.65	4.27	4.46	4.46
BN/Majority Voting	4.03	3.37	4.91	4.14	4.14
BN/Bernoulli	1.83	1.60	2.14	1.87	1.82
BN/TAN	2.38	2.08	2.78	2.43	2.22
BN/multinomial	4.03	1.44	7.48	4.46	5.13
BN/multinomial (MV corr)	2.48	2.08	2.99	2.54	2.56
MLP	1.92	2.08	1.71	1.90	2.11
SVM	2.75	2.72	2.78	2.75	2.75

Table 7.6 — Results of decision-level fusion models on the BANCA database. The statistics are given as an average of over G1 and G2.

7.5.2 Score-level fusion

Experimental setup

For the Sparse Regression Fusion algorithm, we show the best and worst result according to the independence threshold $\bar{I}_\tau$ (see Section 7.4.5).

In the mixture of softmax densities model (logistic regression), we train as many densities as there are classifiers in the ensemble.

The BN/TAN is implemented by discretising continuous variables.

Results and discussion on XM2VTS

The results for fusion experiments on XM2VTS are provided in Table 7.7.

In these experiments, the SRF algorithm consistently outperformed both the single best classifier in the ensemble and the mean rule in terms of EER, except in a few cases where performance was equivalent to that of the mean rule. This can be attributed to the fact that the evaluation dataset available is large enough for the algorithm to be able to pick out meaningful relationships between variables. The fact that performance is systematically better when mixture nodes are included suggests that it beneficial to drop the Gaussianity assumption about scores. Overall, using the best possible independence threshold, the results are the best on the set. The performance is equivalent to the mixture of logistic regressors. Using the worst possible threshold, modelling even the correlations with classifiers that share litteraly no mutual information, results in a performance drop but the algorithm still provides much lower error rates than the baseline best classifier.

Using mixture of logistic regressors (softmax densities) instead of a single logistic regression function lowers the overal error rate, suggesting that this may in fact be an appropriate ensembling strategy.

The strong constraints on the model topology in the BN/TAN model serve it well, and the error rates are equivalent to that of a state-of-the art multilayer perceptron combiner.

fusion classifier	**M**	**err** [%]	**FAR** [%]	**FRR** [%]	**HTER** [%]	**EER** [%]
base best (face)	1	1.040	1.040	1.250	1.140	1.070
BN/SRF ($\bar{I}_\tau = 0.03$)	1	0.6	0.59	1.25	0.92	0.92
BN/SRF ($\bar{I}_\tau = 0.01$)	1	0.65	0.65	1.75	1.20	1.09
BN/SRF ($\bar{I}_\tau = 0.03$)	4	0.01	0.01	1.0	0.51	0.25
BN/SRF ($\bar{I}_\tau = 0.01$)	4	0.02	0.01	1.75	0.88	0.50
BN/logistic regression	1	0.05	0.05	1	0.52	0.30
BN/logistic regression	8	0.01	0.004	1.25	0.63	0.26
BN/CART	1	0.02	0.01	1.50	0.76	0.76
BN/TAN	1	0.02	0.01	2.0	1.0	0.38
mean rule	1	0.38	0.38	0.75	0.56	0.50
MLP	1	0.03	0.03	1.0	0.51	0.38
SVM	1	0.01	0.002	1.25	0.63	0.63

Table 7.7 — Results of score-level fusion models on the XM2VTS database. M denotes the number of classifier components for mixture-based classifiers.

Results and discussion on BMEC 2007

The results for fusion experiments on BMEC 2007 are provided in Table 7.8.

Since the BMEC 2007 ensemble consists of three classifiers each belonging to a different modality, it is expected that the best performing models will take into account the conditional independence between scores. Indeed, in terms of error rates, it can be seen that the SRF model performs better when having higher thresholds for independence, meaning that fewer regressions between scores are modelled. The adjunction of mixture nodes helps for SRF, again confirming that the scores do not follow a Gaussian distribution, and that a low-order mixture (4) helps bring the model closer to the

distribution in the data. The bad performance in terms of FRR suggests that the decision threshold may have to be chosen differently.

The BN/CART model, MLP, and SRF with best threshold all have the same error rate, suggesting a limit on the accuracy of single classifiers that may probably be overcome by using ensembling techniques.

On this database, the use of mixtures of logistic regressors instead of a single logistic regressor provides no significant improvement in terms of error rate, even increasing the EER.

Overall, all combiners bring very sigificant decrease in error rates (up to more than 70% over the baseline best). This is clearly attributable to the fact that this is a trimodal fusion setting, where conditional independence relationships between the classifiers in different modalities hold, ensuring diversity.

fusion classifier	**M**	**err** [%]	**FAR** [%]	**FRR** [%]	**HTER** [%]	**EER** [%]
base best (fingerprint)	1	13.08	13.10	13.00	13.05	13.05
BN/SRF ($\bar{I}_\tau = 0.1$)	1	5.00	1.70	21.50	11.60	7.10
BN/SRF ($\bar{I}_\tau = 0.05$)	1	5.25	1.90	22.00	11.95	7.25
BN/SRF ($\bar{I}_\tau = 0.1$)	4	3.67	1.10	16.50	8.80	7.00
BN/SRF ($\bar{I}_\tau = 0.05$)	4	5.08	2.00	20.00	11.25	6.90
BN/logistic regression	1	3.58	1.30	15.00	8.15	4.55
BN/logistic regression	3	3.42	1.00	15.50	8.25	5.00
BN/CART	1	3.67	2.2	11.0	6.6	7.65
BN/TAN	1	3.75	1.90	13.00	7.45	7.00
mean rule	1	5.83	3.70	16.50	8.55	8.55
MLP	1	3.67	2.10	11.50	6.80	6.00
SVM	1	5.08	0.60	27.50	14.05	14.05

Table 7.8 — Results of score-level fusion models on the BMEC 2007 database. M denotes the number of classifier components for mixture-based classifiers.

Results and discussion on BANCA

The results for fusion on the BANCA database are shown in Table 7.9.

The results of the SRF algorithm indicate that modelling correlations between classifiers within the same modality is important. As can be seen in table Table 7.9, the best result in terms of EER are for a threshold of $\bar{I}_\tau = 0.15$, which is in accordance with Figure 7.21(b). This indicates that the "mutual information mass" drops sharply after significant dependencies are modelled: indeed, all score pairs after the cliff of Figure 7.21(b) are between-modality pairs, while pairs before the cliff are within-modality pairs. The resulting SRF model corresonds to a configuration where all the classifiers for the same modality in the ensemble are grouped under a single mixture node, and no arc exists between modalities. Modelling less dependencies is detrimental, as within-modality correlations are ignored. Likewise, lowering the threshold too much yields an increase in the modelling of spurious dependencies, resulting in an increased error rate. Again, mixture modelling in the SRF model is essential to accomodate the distribution of classifier scores, and results in improved performance.

Except for the SRF model, the combiners based on Bayesian network underperform when compared to the MLP and SVM; however, the mixture of logistic regression model is able to obtain a slightly lower error rate than the SVM.

fusion classifier	**M**	**err** [%]	**FAR** [%]	**FRR** [%]	**HTER** [%]	**EER** [%]
base best (speech)	1	4.49	4.65	4.27	4.46	4.46
BN/SRF ($\bar{I}_\tau = 0.15$)	1	3.57	1.28	6.62	3.95	2.7
BN/SRF ($\bar{I}_\tau = 0.21$)	1	4.67	2.24	7.91	5.07	3.85
BN/SRF ($\bar{I}_\tau = 0.15$)	4	2.2	1.12	3.63	2.38	1.66
BN/SRF ($\bar{I}_\tau = 0.21$)	4	3.85	2.56	5.56	4.06	3.58
BN/logistic regression	1	1.65	1.28	2.14	1.71	1.79
BN/logistic regression	7	1.37	1.12	1.71	1.42	1.10
BN/CART	1	3.11	3.37	2.78	3.07	3.10
BN/TAN	1	2.47	1.76	3.42	2.59	2.38
mean rule	1	15.29	0.0	36.0	17.84	2.72
MLP	1	1.65	1.12	2.35	1.74	1.55
SVM	1	1.47	0.64	2.56	1.60	1.60

Table 7.9 — Results of score-level fusion models on the BANCA database. The statistics are given as an average of over G1 and G2. M denotes the number of classifier components for mixture-based classifiers.

7.6 Summary

In this Chapter we have provided probabilistic interpretation for many decision-level and score-level fusion algorithms, in the first attempt to offer a systematic view of multiple classifier combination using Bayesian networks.

We have shown that arbitrary boolean functions can be realised by Bayesian networks, opening the way to an infite array of novel decision-level fusion functions

We have shown that the topologies of Bayesian-network combination schemes could be divided roughly into generative generative and discriminative approaches, both for decision-level fusion and score-level fusion.

We have proposed probabilistically motivated improvements (majority voting and parameter smoothing) to the multinomial combiner, and shown them to significantly reduce the error rate of this combiner in some datasets, typically where not much data is available and the multinomial combiner is likely to overfit.

In score-level fusion, following the general principle of mixture modelling in Bayesian networks, we have proposed using a mixture of softmax distributions (logistic regressors) to improve on the results of single logistic regression functions. In experiments, the mixture modelling was generally shown to reduce errors.

We also insisted on the idea that in order to properly evaluate the dependencies between classifiers, it is necessary to use conditional mutual information, rather than simple mutual information, in order to take into account real dependencies. Based on this observation, we have proposed a novel structure learning algorithm for multiple classifier combination, which works by modelling "important" independences between classifiers, or conversely by not modelling weak dependences. We have proposed a way to select the most important parameter in this algorithm, namely the independence threshold, by looking at a graph of conditional mutual informations between classifier pairs.

We have provided an analysis of dependencies between classifiers in multimodal and unimodal fusion, and showed numerically the differences of magnitude that can be expected on real data.

Experimental results have shown that on three reference biometric databases, Bayesian-network basd fusion performs at least as well as state-of-the-art methods, in some cases outperforming an MLP and an SVM.

Multiple classifier systems using quality measures

8

8.1 Introduction

As we saw in Chapter 7, using multiple classifier systems allows for the weaknesses of a particular classifier to be somewhat compensated by relying on other ensemble members. In real-world multimodal biometric authentication, where acquisition conditions can be degraded for some modalities, but not for others, this principle is of great importance. While using multiple classifiers is a very important step towards adressing the problem of variability in biometrics, including quality measures in the multiple classifier modelling process allows for even greater gain, as the deficiencies of each modality or classifier can be explicitely taken into account in the fusion process.

However, the majority of work on multiple classifier systems for biometric authentication has generally focused on combining decisions or scores, but not additional information. In this chapter, we propose a theoretical analysis of several important issues arising when combining classifier outputs and quality measures using probabilistic models. We also introduce the concept of *context-specific independence* [35] and its pertinence to the problem at hand. We use the related notion of *relevance* to explain why some classical classifier combination algorithms may fail when used with quality measures.

This chapter is organised as follows: Section 8.2 discusses three fundamental issues related to quality measures that must be taken into account when designing multiple classifier systems using quality measures. Section 8.5 describes a reliability-based scheme for multi-classifier decision fusion with quality measures, and discusses its theoretical limits. Section 8.3 proposes an extension to the Sparse Regression Fusion algorithm presented in Chapter 7 to account for quality information. Section 8.4 introduces the notion of homogeneous context modelling and proposes a fusion model based on maximising independence between random variables. Section 8.6 presents an experimental

evaluation of the three proposed quality-based fusion algorithms, and Section 8.7 summarises the chapter.

8.2 Theoretical issues in quality-dependent combiner design

In this section, we investigate the notions of independence and conditional independence in the context of the use of quality measures, as some theoretical refinements are needed to handle quality measures satisfactorily.

8.2.1 The dangers of univariate modelling

We first show why algorithms judging the merits of a feature by itself, without reference to other features, are likely to fail for our intended application of multiple classifier fusion using quality measures. To this end, following Guyon et al. [114], we define individual feature irrelevance as follows:

Definition 29 (Individual feature irrelevance) *A feature O_d is individually irrelevant to the class Ω iff $O_d \perp\!\!\!\perp \Omega$.*

Hoever, an feature can become relevant if considered together with another feature. We now show an example, closely linked to the case of combining classifier outputs and quality measures, where an individually irrelevant feature is actually useful when combined with another feature. In Figure 8.1, it can be seen that on its own, the quality measure QM is not discriminative with respect to the class, and can therefore be thought irrelevant. That is, $QM \perp\!\!\!\perp \Omega$. However, when considered together with the classifier output score Sc, it can be seen that it is not irrelevant anymore, that is $\{QM, Sc\} \not\perp\!\!\!\perp \Omega$. In fact, the resulting linear discriminant improves over the error rate of the score alone.

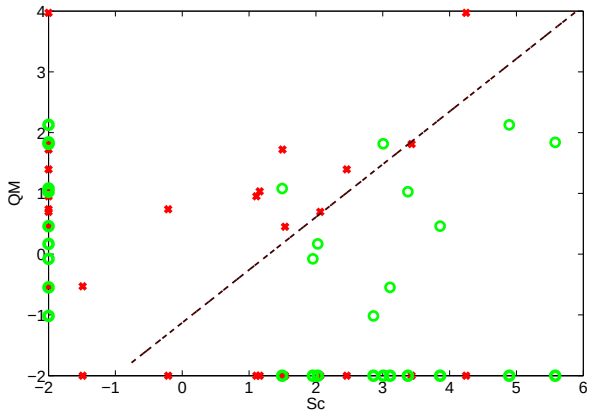


Figure 8.1 — The problem of univariate irrelevance. Crosses represent class 0, circles class 1, and the dashed line is the decision boundary of a linear discriminant function separating the classes. The QM and Sc marginals are shown on their respective axis. The data is synthetic.

This problem is pervasive in the use of quality measures: many pattern recognition algorithms actually resort to univariate computation at some point.

Feature selection search algorithms such as individual rankings will fail, while for example floating search [238] takes into account the inter-relationships between features and thus provides usable

results. Likewise, the general approach of choosing features that have high mutual information with the class will fail.

Classification algorithms that rely on modelling features independently will in general perform poorly on quality-based fusion tasks. For instance, it is expected that a naïve Bayes model (see Section 7.2.1) will not perform well since quality measures by themselves do not carry class-dependent information.

8.2.2 Functional forms of probability densities for quality-based fusion

The five main random variables in quality-based fusion are the classifier output score Sc, the classifier decision CID, the class Ω, the error indicator DR, and the quality measure QM. Depending on the topology of the Bayesian network in which they appear as nodes, different independence relationships will hold.

We first formulate some desirable properties the relationships between these variables should posess. As we will see, it is not obvious to devise a model that respects all of these properties, or that respects them unconditionally.

1. Since the quality measure is a quantity indicative of factors affecting classifier output, it should not be independent from the score or decision. Thus, the first requirement is $QM \not\perp\!\!\!\perp Sc$, $QM \not\perp\!\!\!\perp CID$.

2. Quality measures should be indicative of errors, and thus we have $QM \perp\!\!\!\perp DR$.

3. Modality-dependent quality measures indicate signal degradation, and they should not depend on the class as we cannot *a priori* postulate that noise affects impostors and clients differently*; thus we have $QM \perp\!\!\!\perp \Omega$.

We distinguish three main topologies using these variables, each with different implications in terms of independence assumptions. We show models using scores rather than decisions, without loss of generality.

Generative modelling

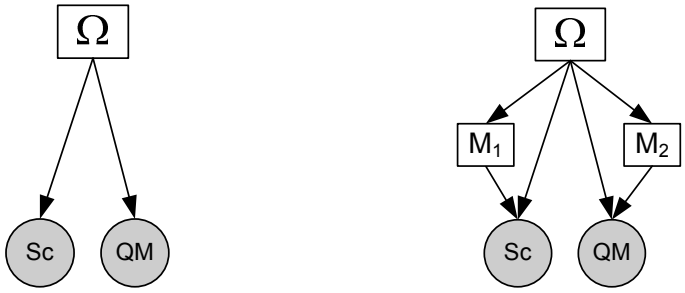
The generative modelling approach puts the class node as the root of the tree. This topology can be used to model various degrees of dependence between random variables (here scores and quality measures), as we have seen in Section 7.4.5. The first possibility is to treat the problem using a naïve Bayes approach. This is shown in Figure 8.2.

In this case, we have $QM \perp\!\!\!\perp Sc|\Omega$, meaning that if we use supervised training (Ω is observed) we learn the two marginal densities (scores and quality measure) separately. Furthermore, we have $QM \not\perp\!\!\!\perp \Omega$, which is contrary to the desiderata formulated above. Accordingly, it seems that the use of generative modelling using the naïve Bayes assumption for combining classifiers and quality measures will not work. In functional terms, the class posterior is

$$P(\Omega|Sc, QM) = \frac{P(\Omega)P(Sc|\Omega)P(QM|\Omega)}{\sum_{\Omega} P(\Omega)P(Sc|\Omega)P(QM|\Omega)}. \tag{8.1}$$

The class-conditional quality measure marginals $P(QM|\Omega)$ have a non-informative contribution to the class posterior, as the corresponding densities have nearly exactly the same parameters for $\Omega = 1$ and $\Omega = 0$. This is illustrated in Figure 8.3

*However, as we have seen in Section 5.4.4, it is important to note that this may be true in the signal domain, but does not necessarily hold in the score domain, depending on the decision rule used.



(a) Using normal distributions (b) Using mixture distributions

Figure 8.2 — Generative modelling of scores and quality measures assuming independence between scores and quality measures.

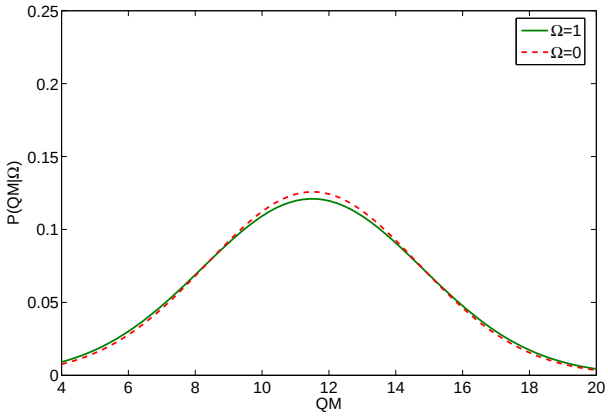


Figure 8.3 — Example class-conditional quality measure marginals on BANCA G1, using the QM_{VAD_E} SNR-related quality measure, and showing non-informativeness of such marginals.

The second possibility is to model explicitely the correlation between quality measures and scores, yielding the topology shown in Figure 8.4*. This is an attempt to correct one major deficiency of the naïve Bayes approach, namely the lack of joint modelling of scores and quality measures.

(a) Using normal distributions (b) Using mixture distributions

Figure 8.4 — Generative modelling of scores and quality measures assuming dependence between scores and quality measures.

In this case, we have $QM \not\perp\!\!\!\perp Sc$, since we learn a regression edge $QM \rightarrow Sc$, meaning the training algorithm will learn the correlation between quality measures and scores explicitely. However, we still have $QM \not\perp\!\!\!\perp \Omega$. In functional terms, the class posterior is

$$P(\Omega|Sc, QM) = \frac{P(\Omega)P(Sc|\Omega, QM)P(QM|\Omega)}{\sum_{\Omega} P(\Omega)P(Sc|\Omega, QM)P(QM|\Omega)}. \tag{8.2}$$

While the class-conditional quality measure marginals still have a non-informative contribution to the class posterior, the $P(Sc|\Omega, QM)$ term allows for taking into account the effect of quality measures on scores. For illustration purposes, we use a discretised version of quality measures, yielding binary quality measures $\widehat{QM}_.$:

$$\begin{aligned} \widehat{QM}_. &= \text{good} \quad \text{if} \quad QM_. \geq \overline{QM}_. \\ \widehat{QM}_. &= \text{bad} \quad \text{if} \quad QM_. < \overline{QM}_., \end{aligned}$$

where $\overline{QM}_.$ represents the average value of the quality measure over the training corpus.

A $P(Sc|\Omega, \widehat{QM}_.)$ class- and quality-conditional score distribution is illustrated in Figure 8.5. It can be seen that this functional form captures the intended information, namely the change in score distribution due to changing acquisition conditions, as reflected by a quality measure. The discretised version of the quality measure acts as a mixing node, performing supervised clustering of the score space into two distributions.

Once these functional building blocks integrating quality measures and scores are formulated, they can be used for fusing multiple classifier, in unimodal or in multimodal systems. These two cases are illustrated in Figure 8.6†, where for simplicity we omit the mixture nodes.

The difference is that each modality in a multimodal system can resort to a different quality measure, while in theory modality-dependent quality measures are shared between classifiers of the same modality. Within one modality, it is even possible that some quality measures are not usable for all classifiers, for instance a brightness-related quality measure may not be useful if the

*Note that by removing the $\Omega \rightarrow QM$ edge we obtain the building block of the model of Baker and Maurer [15], or equivalently the JQS model of Poh et al. [234]

†Note that the generative versions of the fusion models proposed by Kittler et al. [153] can be obtained by selectively removing arcs or making nodes vector-valued. By removing the arcs between the score nodes in Figure 8.6(b) we obtain the generative MSSP model. By making the score nodes vector-valued and removing the arc between score nodes we obtain the generative MSJP model. The generative SSJP model can be obtained by having a single vector-valued score node.

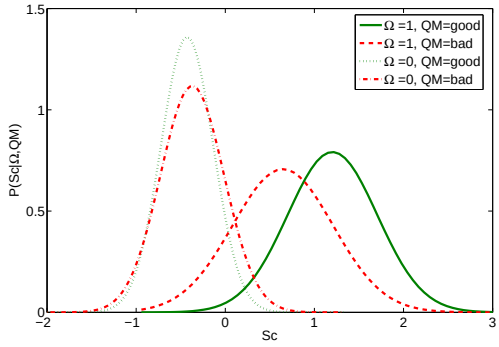


Figure 8.5 — Example class- and quality-measure conditional score marginals on BANCA G1, using the QM_{VAD_E} SNR-related quality measure discretised to two states (*good* and *bad*).

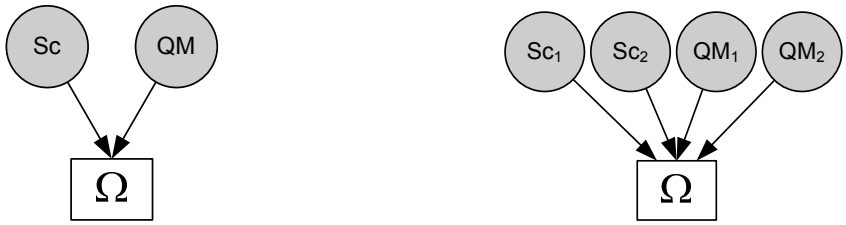
base classifier uses illumination normalisation as a preprocessing step. In this case the model of Figure 8.6(b) can be used.

(a) Unimodal system (b) Multimodal system

Figure 8.6 — Quality-dependent fusion using a generative modelling approach. Dashed arcs represent optional arcs.

Discriminative modelling

The topology corresponding to the discriminative modelling approach is shown in Figure 8.7(a). In this case, we have $QM \not\perp\!\!\!\perp Sc|\Omega$. Since we use supervised training with Ω visible, we learn a joint density for (QM, Sc). However, we also have $QM \not\perp\!\!\!\perp \Omega$, which is contrary to the desiderata formulated above.



(a) Topology for discriminative modelling (b) Example fusion of 2 classifiers

Figure 8.7 — Discriminative modelling of scores and quality measures.

If we use multivariate logistic regression, we can resort to softmax densities for the Ω node, and

the class posterior is:

$$P(\Omega = \omega|\mathbf{SQ}) = \frac{e^{\mathbf{W}'_\omega \mathbf{SQ} + \mathbf{b}_\omega}}{\sum_\omega e^{\mathbf{W}'_\omega \mathbf{SQ} + \mathbf{b}_\omega}}, \tag{8.3}$$

where $\mathbf{SQ}$ is a vector concatenation of Sc and QM

$$\mathbf{SQ} = [\ Sc \quad QM\ ]', \tag{8.4}$$

and the other terms are as per Equation(7.22).

Since regression involves minimising the prediction residuals after solving an overcomplete system of linear equations, and given that quality measures carries no class-discriminant information, their weight in the linear combination of features tends to be very low, typically one or two orders of magnitude below that of scores*. Accordingly, it is expected that this type of model will only bring marginal improvements over trained fusion using scores only. One possible solution to the problem is to use non-linear regression by taking into account cross terms (between scores and quality) in the objective function [304], and another possibility is to apply a feature tranform to pre-fuse quality and scores [153].

This model can easily accomodate several scores and quality measures by redefining

$$\mathbf{SQ} = [\ \mathbf{Sc} \quad \mathbf{QM}\ ]', \tag{8.5}$$

where now both $\mathbf{Sc}$ and $\mathbf{QM}$ are vector-valued. However, doing so reveals another weakness of the regression approach: the handling of multimodal fusion with quality measures is not satisfactory. By solving for the regression coefficients, we cannot take into account the fact that quality measures are only relevant to specific individual score terms in the linear equation. The same problem appears in unimodal fusion with user-model dependent quality measures, as these are tied to a particular classifier's output, not to all classifier outputs.

Causal modelling

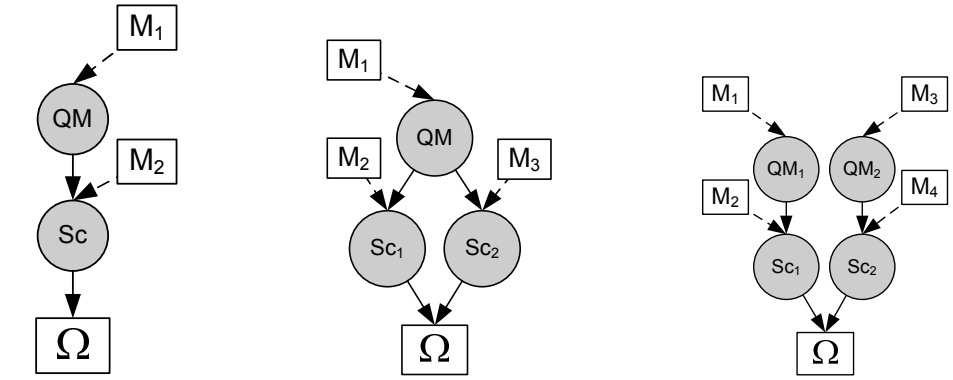
By changing the semantics of having an edge between two nodes $A \rightarrow B$ from "The distribution of random variable B is conditionally dependent on A" (see Section 3.2) to "Random variable A directly causes random variable B", a Bayesian network can be considered as a *causal* Bayesian network [114]. This perspective points to another topology for modelling the relationships between quality measures and scores.

Since quality measures reflect phenomena affecting classifier outputs, it is sensible to say that the phenomena represented by quality measures are causes of observed values for scores, and thus have $QM \rightarrow Sc$. Secondly, in a fusion setting the classification decision is based on observed scores, so a $Sc \rightarrow \Omega$ edge is mandated. The building block for the corresponding fusion model is shown on Figure 8.8(a).

As in the generative model with explicit modelling of correlation between scores and quality measures, this topology entails $QM \not\perp\!\!\!\perp Sc$. It also entails $QM \perp\!\!\!\perp \Omega|Sc$: since scores are always observed, the fact that $QM \not\perp\!\!\!\perp \Omega$ is not relevant. Thus, it seems that this model satisfies all the desirable properties expressed at the beginning of the section.

Taking as an example of application of this topology the multimodal fusion of two classifiers shown in Figure 8.8(c), we also obtain the desirable properties that $QM_1 \perp\!\!\!\perp QM_2$, or more to the point $QM_1 \perp\!\!\!\perp QM_2|Sc_{\{1,2\}}$, which always happens in practice since we deal with observed scores.

*An example on XM2VTS is found in [154], where least mean squares optimisation results in quality weights 3 orders of magnitudes below score weights. This effect is also noted in [153].



(a) Topology for causal modelling

(b) Fusion of 2 classifiers: unimodal system

(c) Fusion of 2 classifiers: multimodal system

Figure 8.8 — Causal modelling of scores and quality measures.

As for the scores themselves, we have $Sc_1 \not\perp\!\!\!\perp Sc_2|\Omega$, where the level of dependence between the quality-conditional scores is learned through regression.

In unimodal fusion (Figure 8.8(b)), assuming the effects measured by QM apply to both classifiers, the topology does not change the independence assumptions since QM is always observed.

Unimodal and multimodal combination of classifiers with quality measures

The most important difference between unimodal (intra-modal) and multimodal fusion is in the link between the quality measures and the rest of the fusion model. For intra-modal fusion, modality-dependent quality measures is the same for each classifier in the ensemble, since the modality is the same for all classifiers. Modality-independent quality measures, however, are associated only with their respective classifier. In multi-modal fusion, the situation is different, and modality-specific quality measures are attached to the classifiers for the corresponding modality. This change however is mostly semantic, and the structure of the network need not change: as long as quality measures, modality-specific or otherwise, are given as real numbers, they can be treated as a continuous random variable and their density can be modelled in the corresponding node.

8.2.3 Context-specific independence in quality-based fusion

The standard definition of conditional independence (see Section 3.2.5) is not always sufficient to account for relationships between variables. This is because a conditional independence relationship holds for *all values* of the variable in the conditioning set. However, it may be the case that variables are only dependent for *certain values* of the variable in the conditioning set. This is the notion of context-specific independence [35].

Definition 30 (Context variable) *A context variable is a variable of the conditioning set.*

Definition 31 (Context) *A context is a particular instantiation (a specific value) of a context variable.*

Formally, context-specific conditional independence between two random variables X and Y for a specific value c of the context variable C, denoted $X \perp\!\!\!\perp Y|C = c$, is defined as:

$$P(X|C = c, Y) = P(X|C = c) \quad \text{when } P(Y, C = c) > 0, \tag{8.6}$$

where C is the context variable, and $C = c$ is the context.

In general, the set of context variables $\mathcal{C}$ can contain several variables.

As an example*, a two-classifiers ensemble consisting of a speech classifier and a face classifier should exhibit higher between-modality dependence in clean conditions (as measured by a hypothetical binary speech quality measure $\widehat{QM}_s = good$) than in noisy acoustic conditions ($\widehat{QM}_s = bad$): the noise decorrelates speech scores Sc_s from face scores Sc_f since the face classifier is immune to acoustic noise. Thus, we have $Sc_s \not\perp\!\!\!\perp Sc_f | \widehat{QM}_s = good$, but $Sc_s \perp\!\!\!\perp Sc_f | \widehat{QM}_s = bad$. An illustration on real data, where no attempt has been made to control face data quality, is shown in Figure 8.9. In this case, the normalised mutual information (see Section 5.4.3) between the speech and face score goes from 0.24 for good speech quality $\widehat{QM}_s = good$ to 0.13 for bad speech quality $\widehat{QM}_s = bad$.

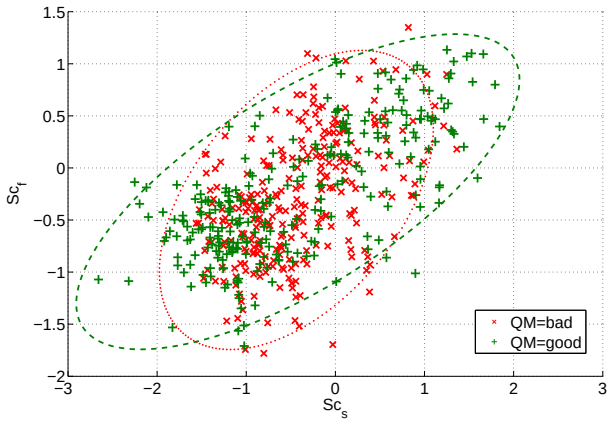


Figure 8.9 — Change in dependency relationship between two classifiers due to degraded speech acquisition conditions, as indicated by a speech quality measure. The plus signs + are the scores for which speech quality is deemed good, while the crosses × are for speech quality deemed bad. The dashed ellipse is the one-standard deviation covariance for good speech conditions, while the dotted ellipse is for bad speech conditions. Sc_s is the score from a speech modality classifier, while Sc_f is the score from a face modality classifier. The dataset is BANCA G1, the quality measure is the binary version of QM_{VAD_E}.

This weaker form of conditional independence is known under several names in the artificial intelligence literature, the most common being *context-specific independence* [35, 330], but other forms such as *contextual weak independence* [315] exist.

It is important to keep this notion in mind when designing fusion models incorporating quality measures: not only does the output distributions of individual classifiers change due to variability in acquisition conditions, but also the dependence relationships between classifiers.

8.3 Sparse regression fusion with quality measures

The sparse regression fusion (SRF) algorithm presented in 7.4.5 can readily be adapted to incorporate quality measures. The simplest option is to incorporate the quality measure nodes as additional random variables, and run the search procedure as described in Algorithm 7.1.

However, SRF tends to overlook connections between scores and quality measures to favour connections between scores of the same modality, since in general the mutual information is larger

*The dataset inspiring this example can be seen in [100, Figure 1], where speech is artificially corrupted by additive noise.

between scores than between scores and quality measures*. Thus, a second option is to use domain knowledge and reduce the search space by force-addition of $QM \rightarrow Sc$ edges to the resulting topology after an SRF search. In this case, we also add edges between the mixture parent of the corresponding score and the quality measure. The resulting topology is an embodiment of the generative functional form depicted in Figure 8.4(b).

As mentioned in Section 8.2.2, the difference between using modality-specific quality measures and modality-independent quality measures is that Sc nodes have common QM parent nodes for modality-specific quality measures, while in the case of modality-independent quality measures such as those presented in Section 5.6, which are classifier-specific, the parent QM variables are unique to each classifier.

8.4 Context-specific fusion models for quality-based classifier combination

In combining multiple classifiers and modelling quality measures, we have observed repeatedly that (in)dependence relationships between random variables can be conditional on other random variables, or even on specific values thereof. For example, in Section 5.4.4 we have shown that the relationship between quality measures QM and scores Sc can depend upon a specific value of the context variable Ω, e.g. we can have $Sc \perp\!\!\!\perp QM|\Omega = 0$, but $Sc \not\perp\!\!\!\perp QM|\Omega = 1$. Also, in Section 7.4.5 (see Table 7.3) we have seen that conditioning on the class variable significantly alters the dependence relationships between classifiers in multimodal fusion. Finally, in Section 8.2.3 we have pointed to the fact that, given a specific value of the quality measure (for example the *context* $\widehat{QM} = bad$), two classifiers might become only weakly dependent, or even modelled as independent.

Thus, it appears that conditional independence relationships, and context-specific independence relationships between class, scores, and quality measures, have an important role to play in quality-based fusion.

To account for these effects, we use the theoretical framework of context-specific independence and the related context-specific Bayesian networks models. We are interested in automating the process of finding sets of context variables $\mathcal{C}$, whose values alter the dependence between other variables in the dataset, in order to build fusion models. We call this method *context-specific fusion* (CSF), and we cast it as a probabilistic interpretation of learning C4.5-type decision trees.

In Section 8.4.1, we start by drawing on previous work showing that conditional probability tables (CPT) in a Bayesian network can be represented as decision trees.

In Section 8.4.2, we extend the classical framework of CPT-as-decision trees by showing how to deal with continuous variables by using discretisation, a process equivalent to that used in decision tree building. We interpret the task of building a decision tree in terms of *conditional* mutual information, instead of the usual approach of using mutual information (gain).

In Section 8.4.3, we propose an interpretation of our method, and of decision tree building, as a way of enforcing a weaker form of context-specific independence.

In Section 8.4.4, based on the notion of conditional independence and context-specic independence, we propose a explanation for the fact that decision trees can actually make use of quality measures to improve fusion.

Section 8.4.5 shows how the existing representation framework for context-specific Bayesian networks can be used implement context-specific fusion, and Section 8.4.6 talks about specific implementation issues.

*Though this is largely data-dependent, we have observed this effect on several databases.

Finally, Section 8.4.7 evokes the potential benefits from ensembling context-specific fusion models.

8.4.1 Representing conditional probability tables as decision trees

As shown by Glesner and Koller [108], Cited in [35], conditional probability tables can be represented as decision trees. For binary variables, it is easily seen that going down the branches of a decision trees is akin to formulating a boolean query on the node tests, which as we have shown in Section 7.3.1 can be implemented as Bayesian networks. The leaves of the decision trees correspond to the probability of the joint event described by the boolean query.

We also showed that the converse operation is possible, namely that transforming a decision tree into a Bayesian network can be done by discretizazion, Section 7.2.4.

An important insight by Boutilier et al. [35] was that decision trees can be used to represent context-specific independence relationships present in conditional probability tables. However, the vast majority of literature published on context-specific Bayesian network algorithms deals only with discrete data ([32, 35, 92, 101]), meaning that the distributions (such as those in Equation (8.15)) are discrete. This is not directly appropriate in our case, as the random variables we are dealing with (scores and quality measures) are continuous. Secondly, context-specific independence as shown in [35] is limited to "local structures", meaning the probability distributions of only one feature is considered. We wish to build models over the whole feature set.

8.4.2 Homogeneous neighbourhoods: dealing with continuous data

Random variables used in probabilistic models, defined on $\mathbb{R}^1$, can be considered as dimensions of a $\mathbb{R}^{D+1}$ feature space. We consider that the class variable $\Omega \in \{0, 1\} \in \mathbb{R}^1$ also consitutes one of the dimensions. This spatial interpretation is a clear indication that in dealing with independence statements, instead of considering discrete random variables only, we can move to a more general form for continuous variables: whereby independence statements hold over discrete domains for discrete variables, they hold over continuous domains in the case of continuous variables.

However, if our goal is to automatically find context variables (Definition 30) whose value allow us to model useful (in)dependence relationships in the data, we cannot use continuous variables directly: the candidate context variables (scores and quality measures) are continuous. As such they can be instantiated to an infinite amount of values, and thus cannot be used directly in context-specific independence (CSI) statements; recall that by the definition of Equation (8.6), CSI is independence holding only for particular values of the context variable.

Thus, one possible approach is to discretise continuous domains into hyperrectangular neighbourhoods and then specify context-specific independence for *discrete values corresponding to specific hyperrectangular neighbourhoods* in the continuous feature space, as opposed to specifying context-specific independence for *specific discrete values* of a discrete random variable.

In this spatial interpretation, a probability distribution defined for a specific value of a (discretised) context variable is defined over points in a specific hyperrectangular neighbourhood in the original feature space. Thus, the task of finding a set of context variables $\mathcal{C}$ can be defined in terms of the effect of the set of context variables on the distribution of a random variable of interest.

Let a feature space consisting of scores, quality measures, and a class variable Ω be defined on $\mathbb{R}^{D+1}$. Let Ω be the variable of interest. Let $\mathcal{C}$ be the set of context variables learned, corresponding to specific discretisations of some dimensions of the feature space, and let $\mathbf{c}$ be the set of values of the context variables in $\mathcal{C}$.

Definition 32 (Homogeneous neighbourhood for Ω) *Hyperrectangular neighbourhood within a feature space $\mathbb{R}^{D+1}$, defined by specific values $\mathbf{c}$ of the context variables in the set $\mathcal{C}$, where the distribution of the variable of interest Ω has the lowest possible conditional entropy given $\mathcal{C}$.*

The search for homogeneous neihbourhoods is a procedure by which the feature space is recursively partinoned into neighbourhoods. Each partition is obtained by looking for the feature whose discretisation gives the most homogeneous neighbourhood. As we will show, this process is equivalent to growing a decision tree using the gain measure [241], which is equivalent to mutual information.

We start with no context variable, and thus an empty set of context variables $\mathcal{C} = \emptyset$. The goal is to find a set of context variables $C_n, n = 1, \ldots, N$ whose values partition the feature space into neighbourhoods as homogeneous as possible.

Assuming we are interested in building a model for the class variable Ω, the first homogeneous neighbourhood, and its associated context variable C_1, is found by:

$$\mathcal{C}(1) = \widehat{C_1} = \underset{C_1}{\operatorname{argmax}}\, I(\Omega; C_1|\emptyset) = \underset{C_1}{\operatorname{argmax}}\, I(\Omega; C_1). \tag{8.7}$$

By the definition of mutual information of Equation (5.1), this is equivalent to minimising the conditional entropy $H(\Omega|C_1)$, since the entropy of $H(\Omega)$ is constant for all contexts and $I(\Omega; C_1) \geq 0$.

However, we want to find discrete context variables so that the concept of CSI can be used, and we want to find homogeneous neighbourhoods. Since in our case the candidate context variables (scores and quality measures) are not discrete, it is necessary to run a search procedure through possible discretisation thresholds, and to use the context variable and discretisation threshold that maximise Equation (8.7). Many discretisation methods are available in classical decision trees litterature. We use the method of Quinlan [240] to transform continuous candidate context variables into discrete binary variables. The first context variable, $C_1 \in \{c_{1_1}, c_{1_2}\}$ is obtained by solving Equation (8.7), and defines two homogeneous neighbourhoods, ν_1 and ν_2, each corresponding to one instantiation of the context variable. Then, ν_1 and ν_2 are recursively divided into sub-neighbourhoods, each time yielding a subsequent context variable. This is achieved by maximising the conditional mutual information Equation (8.8), where the conditioning set is comprised of instantiated context variable C_1. The procedure is repeated, each each time further conditioning the conditional mutual information on the set of context variables selected thus far, $\mathcal{C}(1 \ldots n-1)$, and their instantiation values, which we denote compactly as $\mathbf{c}$.

$$\mathcal{C}(n) = \widehat{C_n} = \underset{C_n}{\operatorname{argmax}}\, I(\Omega; C_n|\mathcal{C}(1 \ldots n-1) = \mathbf{c}). \tag{8.8}$$

Note that in general, decision trees are described using Equation (8.7) (mutual information) only, which is computed within their own neighbourhood. We recognise the importance of past partitions: the fact that we are splitting a particular homogeneous neighbourhood ν_i implies certain specific instantiations of the context variables, which in turn can change conditional independence statements made about the data in the homogeneous neighbourhood.

As per the definition of conditional mutual information of Equation (5.13), solving Equation (8.8) is equivalent to finding $\widehat{C_n}$ such that it has higher conditional mutual information with Ω than all other non-selected candidate context variables C_k:

$$\begin{aligned} \forall k,\, k \neq n, \quad & \underbrace{0 \leq H(\Omega|\mathcal{C}(1 \ldots n-1) = \mathbf{c}) - H(\Omega|C_k, \mathcal{C}(1 \ldots n-1) = \mathbf{c})}_{\text{non-selected candidate context variables}} \\ & < H(\Omega|\mathcal{C}(1 \ldots n-1) = \mathbf{c}) - H(\Omega|\widehat{C_n}, \mathcal{C}(1 \ldots n-1) = \mathbf{c}). \end{aligned} \tag{8.9}$$

We note that, as per Definition 32, Equation (8.9) entails that selected context variable $\widehat{C_n}$ may not make the random variable of interest (Ω in the present case) *independent* of the rest of the dataset; rather, this partitioning of the feature space into homogeneous neighbourhoods corresponds to enforcing a weaker form of context-specific independence, as explained in the next section.

8.4.3 Weak context-specific independence

Algorithms dealing with context-specific independence in Bayesian networks (mentioned in Section 8.4.1) rely in general on a strict definition of context-specific independence (Equation (8.6)), with $X \perp\!\!\!\perp Y|C = c$. This definition is equivalent to saying that for a certain value c of the context variable, knowing the value of Y brings *absolutely no reduction in uncertainty* on the random variable X. This can be expressed in term of conditional entropies:

$$H(X|C = c, Y) = H(X|C = c). \tag{8.10}$$

The strict definition of context-specific independence is further equivalent to saying there is no conditional mutual information between X and Y:

$$\begin{aligned} I(X;Y|C=c) &= H(X|C=c) - H(X|C=c,Y) \\ & \quad H(X|C=c) - H(X|C=c) \\ & \quad 0. \end{aligned} \tag{8.11}$$

As we have seen on real score and quality data in Section 5.7 and Section 7.4.5, even spurious or negligible dependencies can produce non-zero amounts of mutual information. The implementation of the computation of mutual information can be partly responsible for this fact. Thus, in practical problems even independent variable may still have some small amount of mutual information.

Many Bayesian network learning algorithms acknowledge the need for a numerical independence threshold, see for instance [93, 157], or in this thesis Section 7.4.5.

Thus, instead of the *zero-threshold* definition of context-specific independence in terms of conditional mutual information (Equation (8.11)) or the *strict context-specific independence* definition of Equation (8.10), we acknowledge that conditioning X on a random variable Y within context c may in fact slightly reduce conditional entropy, even if X and Y are only very weakly dependent*. Accordingly, we define weak context-specific independence as:

$$H(X|C = c, Y) < H(X|C = c). \tag{8.12}$$

Substituting terms in Equation (8.12) to match Equation (8.9), thus substituting Ω for X, $\mathcal{C}_k$ for Y, and $\mathcal{C}_{n-1} = \mathbf{c}$ for $C = c$ we obtain the definition of weak context-specific independence adapted to the specific case of homogeneous neighbourhoods:

$$H(\Omega|\mathcal{C}(1 \dots n-1) = \mathbf{c}, C_k) < H(\Omega|\mathcal{C}(1 \dots n-1) = \mathbf{c}). \tag{8.13}$$

Then rearranging terms in Equation (8.13):

$$0 < H(\Omega|\mathcal{C}(1 \dots n-1) = \mathbf{c}) - H(\Omega|C_k, \mathcal{C}(1 \dots n-1) = \mathbf{c}). \tag{8.14}$$

Going back to (8.9), we can see that by choosing $\widehat{C_n}$ so as to maximise the mutual information term of Equation (8.8), we necessarily enforce (8.9), meaning non-selected context variables C_k

*This is in accordance with the definition of conditional entropy: conditioning can only reduce entropy [185]

bring a smaller decrease in conditional entropy than the selected context variable $\widehat{C_n}$. In turn, the non-selected candidates term corresponds to the definition of weak context-specific independence of Equation (8.14). Hence, by selecting a context variable that has maximum mutual information with the class (providing a homogeneous context for the class), we ensure that the class is *more* independent independent (not necessarily *strictly* independent) from other non-selected context variables

8.4.4 Individual relevance of quality measures in homogenous contexts

The problems of individual irrelevance mentioned in Section 8.2.1 are avoided by the use of context-specific fusion models.

A feature such as a quality measure may indeed be individually irrelevant to the class *over the whole dataset*, but it may become relevant in a smaller, subsequent homogeneous neighbourhood, because of the already selected context variables.

As we first look for a feature to divide the dataset into two neighbourhoods that are homogeneous for the class variable Ω, by maximising Equation (8.7), a quality measure (or rather, a particular discretisation of it) will not be selected as the first context variable, since its mutual information with the class is very close to zero. By definition, for working classifiers, scores have some non-zero amount of mutual information with the class. Thus, the CSF algorithm will tend to favour scores for the first partition. Assume some discretisazion of score $\widehat{Sc_1}$ has been selected as the first context variable, thus $\mathcal{C}(1) = \{\widehat{Sc_1}\}$

For subsequent subsets, according to Equation (8.8), the conditional mutual information is conditioned on a particular value of the context variable, say $\widehat{Sc_1} = sc_1$. Thus, the computation for QM as a candidate context variable is $I(\Omega; QM|\widehat{Sc_1} = sc_1)$. In this case, it is possible that this quantity is non-null.

This is because, according to the definition of conditional mutual information in terms of joint probabilities in Equation (5.14), two joint probability terms in the conditional mutual information expression, $P(\Omega, QM, \widehat{Sc} = sc_1)$ and $P(QM, \widehat{Sc} = sc_1)$, include both scores and quality measures. Thus, it is possible to have $I(\Omega; QM|\widehat{Sc_1} = sc_1) > I(\Omega; QM)$.

In terms of d-separation, this corresponds to a collider topology, where $\Omega \rightarrow \widehat{Sc} \leftarrow QM$. The implication is that knowing the value of the score renders the quality measure and the class dependent: $\Omega \perp\!\!\!\perp QM$, but $\Omega \perp\!\!\!\perp QM|\widehat{Sc}$.

Thus, algorithms that may at first seem unsuitable for fusing multiple classifiers with quality measures, such as decision trees, may in fact perform well.

Once homogeneous contexts and their related context variables are found, it is possible to implement the context-specific fusion model either as a decision tree or as a Bayesian network. While the implementation as a decision tree is straightforward, the implementation as a Bayesian network requires some additional theoretical background.

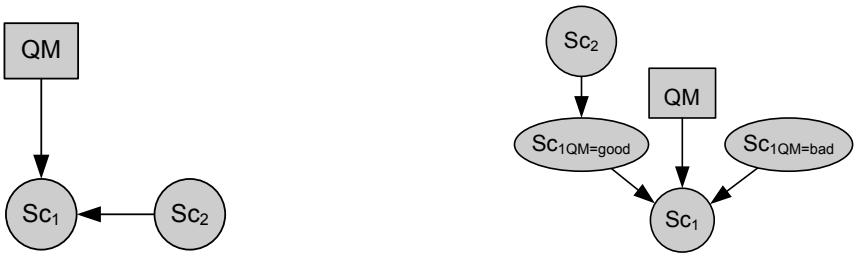
8.4.5 Implementing context-specific fusion models with Bayesian networks

Since context-specific independence is not elegantly supported by the Bayesian networks presented thus far, it is necessary to introduce further refinements to the models presented in Chapter 3.

Boutilier et al. [35] have suggested the use of *multiplexer nodes* in order to implement context-specific independence for discrete random variables, a notion which we adapt for continuous variables. A multiplexer node functions like an ordinary node with an arbitrary number of parents or children, except that it accepts a special kind of discrete parent called a *switching parent*. Depending on the

value of the switching parent, the multiplexer node will be instanciated to the *value* of *only one* of its parents*. Suppose we have a network with the set of edges $\{A \rightarrow C, B \rightarrow C, P \rightarrow C\}$, where P is the switching parent and $\{A, B\}$ are ordinary nodes. When $P = 0$, the value of the C node is equal to the value of A, while in the case $P = 1$, C takes on the value of B. The values of the non-switching parents of the multiplexer nodes represent the value of the multiplexer node *given context* P.

We now illustrate the transformation of a part of a Bayesian network into a context-specific model, corresponding to the situation depicted in Figure 8.9. The original network, not taking into account the independence between classifiers introduced by a low-quality signal on one of the modalities, is depicted in Figure 8.10(a). This network implies the statement $Sc_1 \not\perp\!\!\!\perp Sc_2$, reflecting dependence between Sc_1 and Sc_2. However, this network can only represent the two context-specific relationships of interest, namely $Sc_1 \not\perp\!\!\!\perp Sc_2|\widehat{QM} = good$ (classifiers are correlated when acquisition quality is good) and $Sc_1 \perp\!\!\!\perp Sc_2|\widehat{QM} = bad$ (noise decorrelates modalities) through a specific setting of certain model parameters [105], namely setting regression weights on $Sc_2 \rightarrow Sc_1$ to zero when $\widehat{QM} = bad$.



(a) Original network for multimodal fusion with quality measures on one modality

(b) transformed network encoding context-specific independence between modalities

Figure 8.10 — Modelling of context-specific independence in Bayesian networks using the standard approach (a) and a context-specific approach with a multiplexer node (b). The class nodes are omitted for simplicity.

The transformation of this original network into the context-specific network of Figure 8.10(b) is achieved by splitting the dataset according to the value of the context variable $\widehat{QM} \in \{good, bad\}$. Two conditional distributions can then be learned, corresponding to the two homogeneous neighbourhoods:

$$\begin{aligned} P(Sc_{1QM=bad}) &= P(Sc_1|\widehat{QM} = bad) \\ P(Sc_{1QM=good}) &= P(Sc_1|\widehat{QM} = good, Sc_2). \end{aligned} \tag{8.15}$$

Which distribution is used then depends on the value of the switching parent QM. In the case of good quality, Sc_2 is part of the conditioning set of the Sc_1 distribution, while in the case of bad quality it can be seen that the Sc_1 distribution does not depend on the value of Sc_2 (they are independent). This is equivalent to explicitely setting the regression weight for Sc_2 to zero in the conditional Gaussian distribution of node Sc_1.

*Note that the semantics of a multiplexer nodes is different in the implementations of Murphy [198] and Bilmes and Zweig [23]. In these cases, depending on the value of the switching parent, the multiplexer node will have as parent *only one* of its parents. In the case of a Gaussian multiplexer node, this is equivalent to dynamically setting all but one regression weights on parent arcs to zero. As an example, suppose we have a network with the set of edges $\{A \rightarrow C, B \rightarrow C, P \rightarrow C\}$, where P is the switching parent and $\{A, B\}$ are ordinary nodes. When $P = 0$, the network becomes $\{A \rightarrow C\}$, while in the case $P = 1$, we have $\{B \rightarrow C\}$.

By learning the context-specific model (Figure 8.10(b)) instead of the full model (Figure 8.10(a)), fewer parameters need to be specified (we save learning the regression weight for the conditional mean in the case $\widehat{QM} = bad$). Thus, each parameter has on average more samples available, and its estimate is more robust [92].

Another possibility for implementing context-awareness in Bayesian networks is to use multinets [105, 120], whereby several networks with different topologies are learned for different values of the class variable, and recombined through a class prior. In our application, we could learn several networks for different states of the context variables. This is illustrated in Figure 8.11.

(a) partial network for the context $\widehat{QM} = good$

(b) partial network for the context $\widehat{QM} = bad$

Figure 8.11 — Two partial context-specific networks that can be used in a multinet configuration to represent context-specific independence.

8.4.6 Distribution choice and capacity control for homogeneous neighbourhoods

To model the distribution of the variable of interest (say the class) within a homogeneous neighbourhood, we can either use continuous distributions or discrete distributions. In both cases the parameters are learned via maximum likelihood within each homogeneous neighbourhood.

For continuous distributions we can use multivariate Gaussians, and for discrete distribution a multinomial distribution can be used. If we choose a binomial distribution, the class probability is equivalent to the class probability at the leaf of a decision tree. Training multinomial distributions is more computationally efficient and requires fewer parameters.

While it may seem that we are making the overall Bayesian network model too complex by increasing the number of distributions used to represent interactions in the data, the advantage in terms of inference is that the clique sizes are reduced: thus, while we have more distributions, each of them is smaller, and overall inference will generally be faster.

If homogeneous neighbourhoods contain too few datapoints, the maximum likelihood estimates on which the probability distributions are based will be biased. The first approach to solve this issue is to set a minimum number of datapoints per homogeneous neighbourhood, and to stop the search before the minimum number of points is reached. This is equivalent to setting a stop criterion for the splitting in decision trees.

8.4.7 Context-specific fusion models ensembles

Classical ensembling methods such as boosting [90] or bagging [38] can be applied to train ensembles of CSFs. By setting a low threshold on the minimum number of points in a homogeneous context, (see Section 8.4.6), we can destabilise the model, meaning that the parameter estimates for some probability distributions of small homogeneous neighbourhoods will have high variance with respect

to changes in training data. Thus, by training a certain number of classifiers on randomly selected subsets of the training data (bootstrap samples), we can create a diverse ensemble of CSFs, which can outperform single CSFs. As CSFs are equivalent to decision trees, it can be expected that other ensembling methods might also bring benefits.

An important but somewhat hidden benefit of forming ensembles of CSFs by bagging is that it can contribute to reducing variance due to the choice of the discretisation thresholds for the continuous context variables. Geurts and Wehenkel [106] have shown that bagging models using discretisation thresholds is one of the most effective way of reducing threshold variance, which can contribute to improve classification accuracy.

8.5 Rigged voting schemes for decision-level fusion

In this section, we will show another approach to incorporating quality measures in the fusion process, which can be applied indifferently to intra-modal and multimodal fusion. Namely, instead of learning distributions in a joint space of scores and quality measures for each classifier, we learn a set of L such distributions, one per classifier.

While majority voting is an appealing combining scheme, its optimality depends on several assumptions*, of which we will mention chiefly the fact that it assumes comparable expertise of the ensemble base classifiers. In biometric applications it is often not the case, especially when combining several modalities, with sometimes one or more orders of magnitude of difference between the error rates of the base classifiers. Furthermore, difference of acquisition conditions or in model quality for different users can cause erroneous decisions to be taken by the base classifiers. Therefore, we propose three schemes that train one meta-classifiers on the output of each base classifier as a way to improve on majority voting.

8.5.1 Rigged majority voting

For rigged majority voting (RMV), we train one meta-classifier on the output of each base classifier and its associated quality measures. At fusion time, the base classifier's decision can be "rigged" (overturned or replaced) by that of its meta-classifier. As previously, we denote the base classifier decision by a binary variable CID (0 for impostors, 1 for clients), the reliability classification by a binary variable DR (0 for unreliable, 1 for reliable), and the rigged decision by RD.

Two possible approaches are possible to train the meta-classifier. The first is to train a reliability model such as those developed in Chapter 6, and to use it to give a soft or hard probabilistic weight to the members of the classifier ensemble. In this case the class of interest is DR.

The second approach is to train a classifier such as those presented in this section, for example an SRF-Q classifer or a CSF-Q classifier. Indeed, other classifiers could be used as well, under certain conditions which we detail later. In this case the class of interest is Ω, and we term this meta-model an Ω-classifier.

If using an Ω-classifier as the meta-classifier, we rig the vote of each base classifer by replacing it by that of the meta-classifier. In this case, the accuracy of the meta-classifier is computed for the Ω class. The reason for improvement over the base classifier is that we model quality measures in addition to scores.

If using a reliability model as the meta-classifier, we can estimate, on an instance-by-instance basis, when the base classifier decision decision is likely to be unreliable. In such cases, the vote

*such as independence of ensemble members

can be rigged by inversion*. This can be implemented by the negative exclusive-or function: $RD = \overline{CID \oplus DR}$. In this case, the accuracy of the meta-classifier is computed for the DR class. The improvement to the final fused accuracy comes from the fact that we can predict errors.

In the case of base classifiers with very different error rates (say, an order of magnitude), the RMV scheme does not guarantee that we can outperform the best base classifier. We therefore introduce a variation on the voting scheme by weighting the contributions of individual classifiers.

8.5.2 Weighted rigged majority voting

The second scheme we introduce, weighted rigged majority voting (WRMV) is also based on rigged votes, which is an instance-specific method, but the rigged votes are subsequently weighted by a factor proportional to the accuracy of that classifier's meta-model model. Thus, we also take into account the overall performance of the base classifier on a development set.

Even though the classifiers violate the independence assumption, and the weights may therefore be suboptimal [171, p.124], we set the classifier-specific weights w_l to

$$\sum_{l=1}^{L} w_l = 1, \;\; w_n \propto \frac{acc_l}{1 - acc_l}, \tag{8.16}$$

where the accuracy of each meta-classifier acc_l is computed according to its confusion matrix.

The difference with standard practice for weighted majority voting is that the accuracy used in weighting is not that of the base classifier, but is replaced by the accuracy of the meta-classifier, which is in principle higher. Thus, the weights are dependent on the effectiveness of each meta-classifier. However, since the accuracies of the meta-classifiers may follow the same ordering as the accuracies of the base models, the results may not always differ significantly.

The majority threshold is changed from $\tau \geq \lfloor L/2 \rfloor + 1$ for unweighted majority voting to $\tau > \sum_{N_{worst}} w_L$. Thus, the vote of the worst N meta-classifiers N_{worst} in the ensemble is insufficient to win the vote, and if meta-classifier accuracies are unbalanced the opinion of the most accurate meta-classifier will count much more. N_{worst} can be chosen as $\lfloor L/2 \rfloor + 1$.

8.5.3 Selective rigged majority voting

The selective rigged majority voting scheme (SRMV) operates on the same principle as the confidence gating method used in [272] and the arbitration scheme of [214]: the classifier with the highest confidence† gets to label the sample. The difference in our case is that we are operating on decisions that have been rigged by the meta-classifier before the selection.

Under some conditions (e.g. three classifiers, one of which clearly dominates for most patterns), selective voting can give results very close to weighted rigged majority voting. This is because the weights assigned to the members of the ensemble are proportional to the error rate of their associated meta-classifier.

In the two-classifiers ensemble case, and using a relibaility model as a meta-classifier, using SRMV is equivalent to the tie-breaking scheme we presented for bimodal fusion in [165]: the posterior output $P(DR = 1|evidence)$ is derived from each classifier's reliability models, and the most reliable modality wins. The corresponding decision table is shown in Table 8.1.

*The role of prior probabilities in the inversion process is discussed in [164]

†For reliability models, this is the meta-classifier that has the highest reliability. For Ω-classifiers, we use the the output of the meta-classifier which has the highest posterior probability, in accordance with many classical confidence estimation measures (Section 2.4).

Classifier 1	Classifier 2	Final decision
$CID_1 = 1$	$CID_2 = 1$	1
$CID_1 = 1$	$CID_2 = 0$	1: $if\ P(DR_1 = 1) > P(DR_2 = 1)$, 0: *otherwise*
$CID_1 = 0$	$CID_2 = 1$	1: $ifP(DR_1 = 1) < P(DR_2 = 1)$, 0: *otherwise*
$CID_1 = 0$	$CID_2 = 0$	0

Table 8.1 — Decision table for bimodal decision fusion equivalent to SRMV with a reliability model as meta-classifier.

8.5.4 Accuracy bounds on rigged voting schemes

The accuracy of the rigged voting schemes is dependent upon the accuracy of the meta-classifiers used to perform vote rigging. In this section, we show the links between error rates of the base classifier and error rates of the meta-classifier, first for reliability models, and then for Ω-classifiers.

We also show theoretical bounds on performance improvement due to rigged voting.

Link in error rates between base classifiers and reliability models

Since we use the verification score (measurement-level output) of the base classifier as one of the features for modelling reliability of decisions, the reliability model is dependent on the accuracy of the base classifier. By definition a well-performing base classifiers has a lower density of scores (which correspond to reliable or unreliable decisions) near the decision boundary than a base classifier with a higher error rate.

However, we can guarantee that the reliability classifier will perform better than the base classifier under certain conditions, which we will phrase in terms of confusion matrices (contingency tables). Let us define $\mathbf{B}$ as the confusion matrix of the base classifier, and $\mathbf{R}$ as the confusion matrix of the reliability model. The classes in $\mathbf{B}$, used by the base classifier, are *0—impostor* and *1—client*, while the classes in $\mathbf{R}$, used by the reliability model, are *0—unreliable* and *1—reliable*.

$$\mathbf{B} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \mathbf{R} = \begin{pmatrix} e & f \\ g & h \end{pmatrix}. \tag{8.17}$$

The two confusion matrices are linked by the fact that the reliability model has as class 0 (unreliable) the errors of the base classifier (off-diagonal elements in $\mathbf{B}$), and conversely as class 1 (reliable) the correct decisions of the base classifier (diagonal elements in $\mathbf{B}$):

$$b + c = e + f, \quad a + d = g + h \tag{8.18}$$

The condition for the reliability model to be able to improve on the output of the base classifier is that the reliability model must make less errors than the base classifier, meaning that the sum of the number of base errors considered reliable and the number of base correct decisions considered unreliable must be less than the sum of the base errors. Equivalently, the accuracy of the reliability model must be higher than that of the base classifier. This formulation can be written as in Equation (8.19) and simplified by using Equations (8.18) to obtain Equation (8.21).

$$\frac{e+h}{(e+f)+(g+h)} > \frac{a+d}{(a+d)+(b+c)} \tag{8.19}$$

$$\frac{e+h}{(e+f)+(g+h)} > \frac{g+h}{(g+h)+(e+f)} \tag{8.20}$$

$$e > g. \tag{8.21}$$

Any reliability model whose confusion matrix satisfies the condition expressed in Equation (8.21) is guaranteed to have less errors than the base classifier it models, and to be useful in reducing base classifier error rates, even if the base classifier performs below chance. If, in addition to reducing base errors, we want the reliability model to perform above chance, we can add the condition

$$e + h > f + g. \tag{8.22}$$

Link in error rates between base classifiers and Ω-classifiers

Letting **B** be the confusion matrix of the base classifier, **R** be the confusion matrix of the Ω-classifier (eq:singleClassifierRel:limits:Confmats), then since the definition of class (Ω) is the same for the base classifier, for the meta-classifier to improve on the output of the base classifier we must have by definition and trivially:

$$b + c > f + g. \tag{8.23}$$

Accuracy bounds on RMV

If the meta-classifier models satisfy Eq. (8.21) or eq:singleClassifierRel:limits:omegaErrors, and assuming the correlation between the rigged votes is the same as the correlation between the votes of the base classifiers, this scheme guarantees better lower and upper bounds on the achievable fused accuracy than simple majority voting on the base classifiers, because the rigged decisions will have higher individual accuracies.

Formally, we draw on the proof by Matan [188], which showed that for an ensemble of L classifiers, the upper and lower bounds on achievable majority voting accuracy are given by:

$$acc_{max} = min(1, f(\tau), f(\tau-1), \ldots, f(1)), \tag{8.24}$$

$$acc_{min} = max(0, g(\tau), g(\tau-1), \ldots, g(1)), \tag{8.25}$$

where the functions $f(\tau)$ and $g(\tau)$ are defined in terms of a specific majority decision threshold τ' (an integer) and base classifier accuracies (acc_l, Equation (2.6)):

$$f(\tau') = \frac{1}{\tau'} \sum_{l=1}^{L-\tau+\tau'} acc_l. \tag{8.26}$$

$$g(\tau') = \frac{1}{\tau'} \sum_{\tau-\tau'+1}^{L} acc_l - \frac{L-\tau}{\tau'}. \tag{8.27}$$

Since both the $f(\tau')$ and $g(\tau')$ functions are linear and increasing in acc_l, by improving the base classifier accuracies both bounds are improved (within $[0, 1]$).

Figure 8.12 shows an example for fusion of three classifiers. It can be seen that for accuracy improvements of about 30% (due to vote rigging), it is possible to obtain perfect classification (upper bound at 1), and to have lower bound no worse than the worse classifier in the baseline ensemble. Depending on the problem setting (database and modality, features, classifier type and complexity, reliability model used), this figure is not unrealistic*.

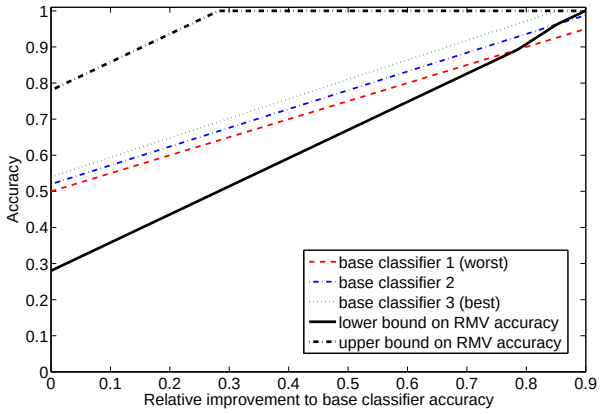


Figure 8.12 — Change in upper and lower bounds of majority voting accuracy as a function of the relative improvement to the accuracies of base classifiers due to rigged votes.

8.6 Experiments and results

The goal of these experiments is twofold. First, to see if and when gains in classification accuracy can be obtained by using quality measures in the fusion process, using the three fusion models proposed in this chapter, for both modality-independent and modality-specific quality measures. Second, the goal is to model the same quality measures with two state-of-the-art fusion algorithms, support vector machines and multilayer perceptrons, and see if they, too, can benefit from the introduction of quality measures in the fusion process.

For intramodal fusion, we use the BMEC 2007 signature database, with one local and one global classifier. The local classifier is a BN/GMM model using B-spline preprocessing, no rotation normalisation, and 36 mixture components for the user model. The features are $(x, y) + \Delta + \Delta\Delta$. The global classifier is also a BN/GMM model using linear interpolation preprocessing, no rotation normalisation, 2 mixture components for the user model, and 11 global features. For both classifiers, we use the $\overline{QM}_{det_w}$ quality measure (Section 5.6.2, Equation (5.39)). We use a 5-fold cross-validation protocol.

For multimodal fusion, we use the BANCA database. with one speech and one face classifier. The speech classifier is the one described in Section 4.4.5. The face classifier is from the IDIAP BANCA database of scores, and is the Surrey neural network classifier. The quality measure used for the speech classifier is QM_{VAD_E} (see Section 5.5.1).

The two trained classifiers we use for fusion are the same as in the experiments of Chapter 7.

*Indeed, using reliability models as meta-classifiers, we have shown it to be possible to achieve up to about 50% relative improvement in error rates on signature, and up to about 40% on speech [257].

8.6.1 Score-level fusion

Intramodal fusion

In intramodal fusion on signature data (results in Table 8.2), we note that, even without quality measures, all fusion models manage to significantly improve the results over the best baseline classifier. This can be attributed to the fact that the two classifiers in the ensemble use different preprocessing, different features, and different model orders, thus yielding good diversity of outputs. In fact, the difference in preprocessing seems to be the most important factor in ensembling signature classifiers.

The use of a quality measure contributes to further improving accuracy, yielding up to 32% improvement over the best baseline classifier for CSF-Q, and up to 39% improvement for the ensemble version of CSF-Q.

The SRF-Q model brings only very marginal improvements in this setting, meaning that the generative topology is not the most appropriate for the type of quality measure at hand.

The CSF-Q and its bagging variant both perform well, as they can define complex decision boundaries, and are discriminative models. This is also the reason why the MLP-Q can take advantage of the quality feature, which defines a non-linearly separable distribution of scores.

The SVM-Q however cannot make use of the quality measure to improve separation of classes, and indeed yields worse results than the SVM operating on scores. This may be due to its use of first-order polynomial kernels or a poor choice of penalty term.

fusion classifier	**M/L**	**err** [%]	**FAR** [%]	**FRR** [%]	**HTER** [%]	**EER** [%]
base best (local)	1	15.09	15.10	15.07	15.08	15.08
SRF ($\bar{I}_\tau = 0.15$)	10	12.63	15.40	8.93	12.17	12.68
SRF-Q ($\bar{I}_\tau = 0.15$)	10	11.71	14.70	7.73	11.22	12.52
CSF	1	10.97	9.80	12.53	11.17	10.85
CSF-Q	1	10.11	8.50	12.27	10.38	10.17
CSF-Q	101	9.66	8.30	11.47	9.88	9.20
mean rule	1	11.31	11.40	11.20	11.30	11.30
MLP	1	10.86	8.30	14.27	11.28	11.08
MLP-Q	1	10.86	8.60	13.87	11.23	10.68
SVM	1	13.71	18.90	6.80	12.85	12.85
SVM-Q	1	14.00	19.00	7.33	13.17	13.17

Table 8.2 — Results on fusing a local and a global classifier at score-level with quality measures on the BMEC2007 database. M denotes the number of classifiers components for mixture-based classifiers, and L denotes the number of base classifiers in an ensemble. The algorithms postfixed with "-Q" use quality measures.

Multimodal fusion

For multimodal fusion of speaker and face verification classifiers (results in Table 8.3), again, all fusion models manage to significantly improve the results over the best baseline classifier. This time, the diversity is ensured by having two different modalities in the ensemble.

The use of a modality-specific quality measure improves over the fusion without quality, yielding reductions over the best baseline classifier of up to 47% in the case of SRF-Q.

The SRF-Q model takes advantage of the relatively high correlation between client scores and the quality measure (Table 5.5) to yield a better model of score distributions.

The CSF model is also improved by the addition of a quality measure, but CSF-Q performs better in an ensemble configuration.

While the MLP is minimally improved by the addition of a quality measure, the SVM achieves very good results when including such information.

fusion classifier	**M/L**	**err** [%]	**FAR** [%]	**FRR** [%]	**HTER** [%]	**EER** [%]
base best (speech)	1	8.06	8.17	7.91	8.04	8.04
SRF ($\bar{I}_\tau = 0.15$)	4	6.59	1.92	12.82	7.37	5.77
SRF-Q ($\bar{I}_\tau = 0.15$)	4	5.68	1.60	11.11	6.36	4.22
CSF	1	6.41	7.21	5.34	6.28	6.12
CSF-Q	1	5.49	5.45	5.56	5.50	5.66
CSF-Q	101	5.22	4.33	6.41	5.37	5.32
mean rule	1	7.14	7.05	7.26	7.16	7.13
MLP	1	5.95	4.33	8.12	6.22	5.13
MLP-Q	1	4.40	3.04	6.20	4.62	4.94
SVM	1	6.32	2.08	11.97	7.02	7.02
SVM-Q	1	4.49	2.08	7.69	4.89	4.89

Table 8.3 — Results of bimodal score-level fusion with quality measures on the BANCA database. The statistics are given as an average of over G1 and G2. M denotes the number of classifiers components for mixture-based classifiers, and L denotes the number of base classifiers in an ensemble. The algorithms postfixed with "-Q" use quality measures.

8.6.2 Decision-level fusion

We used rigged majority voting to test intra-modal fusion in signature verification.

In addition to the local and global signature classifier used in the experiments above, we use a second global BN/GMM classifier. It uses linear interpolation with pen-up interpolation, rotation normalisation, and 5 global features. The model uses 2 Gaussian components with diagonal covariance matrices.

We the two user model-based quality measures in their three variants (Section 5.6.2), resulting in a feature space of 6 quality measures and one score for each meta-classifier.

The meta-classifier trained on the output of each classifier is the CSF-Q model described in Section 8.4. In other experiments, we have used a reliability model as meta classifier [257].

The results in Table 8.4 confirm the effectiveness of RMV over simple majority voting, and over trained rules using scores only. However, as shown by the results of MLP-Q fusion, modelling classifiers independently of each other and their quality measure leads to suboptimal results. A similar result was found in [153].

This is another incarnation of the 'early fusion' paradigm (see Section 2.6.1), which itself may be an seen as a result of the data processing processing theorem [185] in information theory: processing can only destroy information.

8.7 Summary

There is a need for caution when dealing with quality measure for fusion: some algorithms that perform very well in other areas of pattern recognition will not yield satisfactory results when

fusion classifier	**M/L**	**err** [%]	**FAR** [%]	**FRR** [%]	**HTER** [%]	**EER** [%]
base best (local)	1	15.09	15.10	15.07	15.08	15.08
RMV	1	9.49	9.10	10.00	9.55	9.55
MV	1	13.03	15.00	10.40	12.70	12.70
mean rule	1	12.40	12.40	12.40	12.40	12.17
MLP	1	9.66	8.90	10.67	9.78	9.72
MLP-Q	1	6.40	5.50	7.60	6.55	6.00

Table 8.4 — Results of intramodal decision-level fusion with quality measures on the BMEC database. M denotes the number of classifiers components for mixture-based classifiers, and L denotes the number of base classifiers in an ensemble. The algorithms postfixed with "-Q" use quality measures.

applied to this task. The notion of individual feature irrelevance is of great import in this respect, as it provides a theoretical foundation to the use of quality measures.

By analysing the structure of combination models through the use of notions such as conditional independence and d-separation, insights can be gained into why some probabilistic functional forms may perform worse than others. We distinguish three families of functional forms, each with their strength and weaknesses in respect to the ideal requirements we posit. We also mentioned that modality-specific and modality-independent quality measures must be handled differently in unimodal and multimodal contexts: thus, some domain expertise can be combined with a data-driven approach to obtain better fusion models.

One of the most important theoretical point we mentioned was the necessity of taking into account context-specific independence (CSI) when designing fusion models with quality measures: for example noise can decorrelate modalities, but CSI can be observed under many different incarnations when dealing with quality measures. This is the motivation behind the proposition of context-specific fusion (CSF) models, which are a probabilistic model that can be interpreted and implemented as a decision tree (and vice-versa). We showed that maximising the mutual information (gain) with one specific context variable is equivalent to enforcing a weak version of context-specific independence with respect to other variables. The fact that, within a context provided by a specific instantiation of discretised score variables, a quality measure can be modelled jointly with the score, provides an explanation for the reason why CSF is able to take advantage of quality measures.

Furthermore, due to their theoretical equivalence with decision trees, bagging is an attractive way of forming ensembles of CSF models.

We also proposed an extension to the SRF structure learning algorithm of Chapter 7.4 to incorporate quality measures, taking into account the differences between unimodal and multimodal contexts, as well as modality-specific and modality-independent quality measures.

A third model, rigged majority voting and its variants, was proposed as a combination scheme based on improving the results of single classifiers by the inclusion of quality measures, either using the reliability models presented in Chapter 6, or directly modelling for the class variable. We showed the theoretical accuracy bounds of such a combination scheme.

Experiments showed that, for most trained fusion models, the incorporation of quality measures can help lower the error rate compared to a fusion model not using quality measure. The CSF-Q model proved a very good performer for a modality-independent quality measure, while the SRF-F model performed particularly well on a modality-specific quality measure. In both cases, bagging the CSF-Q model improved over the results of the base CSF-Q model.

Furthermore, results indicate that modelling all classifiers jointly with their quality measure is a more effective combination scheme than modelling each classifier independently, then combining

the outputs.

9 Conclusions

In this thesis we proposed to use probabilistic models based on Bayesian networks for both base classifiers and fusion classifiers. We showed that the models developped were general enough to apply to both signature and speech modelling. We proposed the use of Gaussian mixture models for signature verification, implemented as Bayesian networks, and showed that results were equivalent to state-of-the-art signature verification systems.

We then proposed the use of quality measures as additional information to be used in both single-classifier contexts and multi-classifier contexts. We defined precisely the concept of quality measure, and showed the different potential types of quality measures. We proposed new quality measures for both speech and signature, and we proposed the concept of modality-independent quality measure as an additional type of quality measures. We showed that the effect of signal degradation could be different on impostor and client score distributions, an important effect to take into account when designing quality-based fusion models. We proposed a principled evaluation methodology for quality measures.

The use of reliability models was introduced. They are probabilistic models of single-classifier behaviour, taking into account quality measures. The aim was to obtain an enhanced confidence measure, which is to some degree robust with respect to changing environments. Experiments showed that reliability estimation generally outperforms confidence estimation.

We formalised different classifier combination algorithms as probabilistic models in the framework of Bayesian networks for both decision-level and score-level fusion, and proposed enhancements to existing models. We also proposed a new structure learning algorithm, sparse regression fusion (SRF), specifically designed for classifier combination tasks. The SRF model obtained very good results over three multimodal benchmark databases.

Lastly, we proposed a theoretical view on probabilistic classifier combination with quality measure, based on an analysis of independence and conditional independence relationships induced by different model topologies. We also showed the importance of the notion of context-specific independence, and drew a parallel between decision tree building and enforcing a weak version of context-specific independence. Three quality-based fusion schemes were proposed: SRF-Q, an adaptation of the SRF algorithm to the use of quality measures, CSF-Q, a fusion model equivalent

to decision trees but motivated by probabilistic and independence arguments, and rigged majority voting, a flexible scheme that can be used with both reliability models and other meta-classifiers, with clear limits on accuracy gains that can be expected.

9.1 Unimodal biometric verification with Bayesian networks

A Bayesian network topology, equivalent to a GMM, can be used for modelling signatures. The same topology can be used for speaker verification, with the difference of background model adaptation, which we do not perform in our signature verification model.

Furthermore, the same model can be used for modelling both local and global signature features, by reducing the number of Gaussians in the model. In the past, global features have generally not been modelled using the same model families as for local features. While global features generally offer inferior performance, their use can be key to increasing diversity in an ensemble of signature verification classifiers. Likewise, the different preprocessing techniques discussed have a knock-on effect on the rest of the feature extraction chain, and are therefore another effective way to increase diversity without resorting to random subspaces or random sampling methods.

The proposed Bayesian network performs equivalently to a state-of the art approach based on Hidden markov models, which can be implemented as dynamic Bayesian networks. This highlight the point that the temporal aspect of signatures may be less important than the distribution of feature vectors.

9.2 Quality measures in biometric verification

We proposed to divide quality measures into modality-specific and modality-independent. Modality-specific mesures are those which depend directly on the signal, while modality-independent measures do not, but are tied to a particular classifier. Introducing the concept of modality-independent quality measure, we have proposed two related measures of user model quality for probabilistic models, based on properties of the covariance matrix.

Depending on their intended use (error prediction or fusion), these quality measures can be evaluated in several ways. We have pointed out the deficiencies of assuming linear and homoscedastic distributions of scores: Normalised mutual information was proposed for evaluation of quality measures.

We showed that the influence of noise, materialised as a linear shift in the likelihood domain, can be different on client and impostor scores; thus, class-specific evaluation of quality measures is needed, and the normalised conditional mutual information constitutes a useful tool in evaluating the class-specific effect of the quantity measured by quality measures. Furthermore, fusion models incorporating quality measures and scores should pay attention to the fact that the effect of the quality measures may depende on the class: some model topologies would likely not be able to reap the benefits of quality measures if that fact is not taken into account.

The evaluation methods presented can serve as the basis for selecting quality measures when designing quality-dependent algorithms in biometric authentication. The practitioner should however be mindful that, as is the case in feature selection, the best proof of usefulness is obtained by using real classifiers.

In speech, we proposed several quality measures, based on segmentation in the time-domain, and based on higher-order statistics, and showed their good correlation with real signal-to-noise ratio on an artificially corrupted speech database.

Evaluation of quality measures on reference databases showed that both modality-specific and modality-independent quality measures contain information about classifier output scores, thus motivating the quest for quality-based algorithms in biometrics.

9.3 Estimating reliability in single-classifier verification

We proposed to learn a probabilistic model of classifier errors, including score and quality modelling, in the form of a Bayesian network. The effect of various level of quality, as measured by the quality measure, is to change the form of the posterior distribution, thus meaning that the reliability of classification is dynamically computed according to the quality measure. We proposed to use the output of reliability models for either human examination or automated post-processing.

Contrary to many existing confidence measures, we make minimal assumptions about the form of the distributions of scores and quality measures, and use mixture models. The performance of reliability and confidence measure can be assessed by using DET curves, but it is important to take into account the "double-imbalance" problem in confidence and reliability modelling: there is generally less client data than impostor data in the training set, and there are less errors than correct decisions.

Reliability modelling outperforms or at least perform as well as state-of-the-art measures, while offering additional interpretability and flexibility in parameter setting.

9.4 Bayesian networks for combining multiple classifiers

We have provided probabilistic interpretation for many decision-level and score-level fusion algorithms, in the first attempt to offer a systematic view of multiple classifier combination using Bayesian networks.

Bayesian networks are ideally suited to the task of fusing multiple classifiers, as they can be used to implement both generative and discriminative modelling. Furthermore, they can realise arbitrary boolean functions, which suggests that many novel decision-level fusion functions can be implemented using Bayesian networks, while retaining a probabilistic interpretation in terms of multinomial probabilities.

Probabilistically motivated improvements (majority voting and parameter smoothing) can be used to improve the multinomial combiner; they significantly reduce the error rate of this combiner in some datasets, typically where not much data is available and the multinomial combiner is likely to overfit. Using a mixture of softmax distributions generally reduce errors over using a single softmax density.

We have proposed a novel structure learning algorithm for multiple classifier combination, sparse regression fusion (SRF), which works by modelling "important" independences between classifiers, or conversely by not modelling weak dependences. This algorithm is based on the measure of conditional mutual information, rather than simple mutual information, in order to take into account real dependencies. We have shown empirically that dependencies between classifiers are different within-modality and between-modality. As can be expected, multimodality is a good tool to obtain diverse ensembles. The SRF algorithm takes into account the disctinction in a data-driven manner.

Experimental results have shown that on three reference biometric databases, Bayesian-network basd fusion performs at least as well as state-of-the-art methods, in some cases outperforming an MLP and an SVM. The SRF algorithm is a good performer on all databases.

9.5 Multiple classifier systems using quality measures

In general, we found that the theory used in multiple classifier fusion must be refined somewhat to deal with quality measures, as they are not class-discriminative features. In this respect, individual feature irrelevance is an important concept providing theoretical foundation to the use of quality measures.

By analysing the structure of combination models through the use of notions such as conditional independence and d-separation, insights can be gained into why some probabilistic functional forms may perform worse than others. We distinguish three families of functional forms, each with their strength and weaknesses in respect to the ideal requirements we posit. We also mentioned that modality-specific and modality-independent quality measures must be handled differently in unimodal and multimodal contexts: thus, some domain expertise can be combined with a data-driven approach to obtain better fusion models.

We proposed three models for fusing multiple classifier using quality measures: context-specific fusion CSF, which is equivalent to a decision tree, SRF-Q, an extension of SRF for quality measures, and Rigged majority voting.

One of the most important theoretical point we mentioned was the necessity of taking into account context-specific independence (CSI) when designing fusion models with quality measures: for example noise can decorrelate modalities, but CSI can be observed under many different incarnations when dealing with quality measures. This is the motivation behind the proposition of context-specific fusion (CSF) models, which are a probabilistic model that can be interpreted and implemented as a decision tree (and vice-versa). We showed that maximising the mutual information (gain) with one specific context variable is equivalent to enforcing a weak version of context-specific independence with respect to other variables. The fact that, within a context provided by a specific instantiation of discretised score variables, a quality measure can be modelled jointly with the score, provides an explanation for the reason why CSF is able to take advantage of quality measures. Due to their theoretical equivalence with decision trees, bagging is an attractive way of forming ensembles of CSF models, and indeed generally improves performance.

We also proposed an extension to the SRF structure learning algorithm of Chapter 7.4 to incorporate quality measures, taking into account the differences between unimodal and multimodal contexts, as well as modality-specific and modality-independent quality measures.

A third model, rigged majority voting and its variants, was proposed as a combination scheme based on improving the results of single classifiers by the inclusion of quality measures, either using the reliability models presented in Chapter 6, or directly modelling for the class variable. We showed the theoretical accuracy bounds of such a combination scheme.

Experiments showed that, for most trained fusion models, the incorporation of quality measures can help lower the error rate compared to a fusion model not using quality measure. The CSF-Q model proved a very good performer for a modality-independent quality measure, while the SRF-F model performed particularly well on a modality-specific quality measure. In both cases, bagging the CSF-Q model improved over the results of the base CSF-Q model.

Furthermore, results indicate that modelling all classifiers jointly with their quality measure is a more effective combination scheme than modelling each classifier independently, then combining the outputs, thus confirming results by other researchers in the field.

All three fusion models models proposed can be used indifferently for unimodal and multimodal fusion.

9.6 Future directions

The use of objective criteria to evaluate quality measures could be taken further, and using a structure learning algorithm could use them to as cost function to guide a search through possible model topologies.

In the sparse regression fusion algorithm, it should be possible to automate the choice of an independence threshold, probably by computing the gradient of the cumulative mass of mutual information taken into account as further arcs are added between classifier outputs.

In some cases, such as the XM2VTS database, the error rates are so low that it is difficult to establish statistically significant differences between well-performing classifiers. It is likely that the upcoming BioSecure multimodal database would be an interesting proving ground for the algorihtms described in this thesis, especially as the data is collected in various conditions.

Using larger biometric databases is likely to require porting the existing codebase, currently mostly coded in Matlab and Python, to a faster implementation of Bayesian networks such as Intel's PNL, which offers a C++ library.

A very interesting development would be to look at the possible interactions between quality measures and state-of-the art ensembling methods such as AdaBoost, MultiBoost, random forests, or rotation forests. It is possible that the quality measure information could be used to provide an additional criterion in the weighting or deweighting of incorrectly classified examples.

Bibliography

[1] Silvia Acid, Luis M. de Campos, Juan M. Fernandez-Luna, Susana Rodriguez, Jose Maria Rodriguez, and Jose Luis Salcedo. A comparison of learning algorithms for bayesian networks: a case study based on data from an emergency medical service. *Artificial Intelligence in Medicine*, 30(3):215–232, March 2004.

[2] A.G. Adami, R. Mihaescu, D.A. Reynolds, and J.J. Godfrey. Modeling prosodic dynamics for speaker recognition. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, volume 4, pages 788–791, 2003.

[3] A. Adler and T. Dembinsky. Human vs. automatic measurement of biometric sample quality. In *Proc. Canadian Conf. on Electrical and Computer Engineering*, pages 2090–2093, 2006.

[4] S.M. Aji and R.J. McEliece. The generalized distributive law. *IEEE Trans. on Information Theory*, 46(2):325–343, 2000. ISSN 0018-9448.

[5] H. Akaike. A new look at the statistical model identification. *IEEE Trans. on Automatic Control*, 19(6):716–723, Dec 1974.

[6] E Alpaydin and C Kaynak. Cascading classifiers. *Kybernetika*, 34(4):369–374, 1998.

[7] Hakan Altincay and Mubeccel Demirekler. Undesirable effects of output normalization in multiple classifier systems. *Pattern Recognition Letters*, 24(9-10):1163–1170, June 2003.

[8] Howard Anton and Chris Rorres. *Elementary Linear Algebra*. Wiley, 7th edition, 1994.

[9] M. Arcienega and A. Drygajlo. A bayesian network approach for combining pitch and reliable spectral envelope features for robust speaker verification. In *Proc. AVBPA'03*, Guildford, UK, 2003.

[10] M. Arcienega and A. Drygajlo. On the number of Gaussian components in a mixture: an application to speaker verification tasks. In *Proc. Eurospeech 2003*, pages 2673–2676, Geneva, Switzerland, Sept. 2003.

[11] A.M. Ariyaeeinia and P. Sivakumaran. Comparison of VQ and DTW classifiers for speaker verification. In *Proceedings European Conference on Security and Detection (ECOS'97)*, pages 142–146, 1997.

[12] Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1-3):42–54, January 2000.

[13] Kunihiro Baba, Ritei Shibata, and Masaaki Sibuya. Partial correlation and conditional correlation as measures of conditional independence. *Australian and New Zealand Journal of Statistics*, 46(4):657–664, December 2004. doi: 10.1111/j.1467-842X.2004.00360.x.

[14] Enrique Bailly-Bailliére, Samy Bengio, Frédéric Bimbot, Miroslav Hamouz, Josef Kittler, Johnny Mariéthoz, Jiri Matas, Kieron Messer, Vlad Popovici, Fabienne Porée, Belen Ruiz, and (Jean-Philippe) Thiran. The BANCA database and evaluation protocol. In J. Kittler and M.S. Nixon, editors, *Proc. 4th Int. Conf. on Audio- and Video-Based Biometric Person Authentication (AVBPA)*, volume LNCS 2688, pages 625–638, 2003.

[15] John P. Baker and Donald E. Maurer. Fusion of biometric data with quality estimates via a bayesian belief network. In *Biometric Consortium Conference*, Arlington, USA, 2005.

[16] Ricardo Barandela, Rosa M. Valdovinos, J. Salvador Sánchez, and Francesc J. Ferri. The imbalanced training sample problem: Under or over sampling? In *Proc. SSPR & SPR 2004*, volume 3138 of *LNCS*, pages 806–814. Springer-Verlag, January 2004.

[17] David Barber. Machine learning: a probabilistic approach. (lecture notes), 2006.

[18] S. Bengio, J. Mariéthoz, and M. Keller. The expected performance curve. In *International Conference on Machine Learning, ICML, Workshop on ROC Analysis in Machine Learning*, 2005.

[19] Samy Bengio and Johnny Mariéthoz. The expected performance curve: a new assessment measure for person authentication. In *Proc. ODYSSEY 2004 - The Speaker and Language Recognition Workshop*, pages 279–284, 2004.

[20] Samy Bengio, Christine Marcel, Sebastien Marcel, and Johnny Mariethoz. Confidence measures for multimodal identity verification. *Information Fusion*, 3(4):267–276, December 2002.

[21] Roman Bertolami, Matthias Zimmermann, and Horst Bunke. Rejection strategies for offline handwritten text line recognition. *Pattern Recognition Letters*, In Press, Corrected Proof:–, 2006. doi: 10.1016/j.patrec.2006.06.002.

[22] J. Bigun, J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez. Multimodal biometric authentication using quality signals in mobile communications. In *Proc. 12th Int. Conf. on Image Analysis and Processing*, pages 2–11, 2003.

[23] J. Bilmes and G. Zweig. The graphical models toolkit: an open source software system for speech and time-series processing. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages IV–3916–IV–3919vol.4, 13-17 May 2002. doi: 10.1109/ICASSP.2002.1004774.

[24] J.A. Bilmes. Factored sparse inverse covariance matrices. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 1009–1012, 2000. doi: 10.1109/ICASSP.2000.859133.

[25] Jeff A. Bilmes and Katrin Kirchhoff. Directed graphical models of classifier combination: application to phone recognition. In *Proc. 6th Int. Conf. on Spoken Language Processing (ICSLP 2000)*, volume 3, Beijing, China, October 2000.

[26] F. Bimbot, G. Gravier, J.-F. Bonastre, C. Fredouille, S. Meignier, T. Merlin, I. Magrin-Chagnolleau, J. Ortega-García, D. Petrovska-Delacrétaz, and D.A. Reynolds. A tutorial on text-independent speaker verification. *Eurasip Journal on Applied Signal Processing*, 2004(4): 430–451, 2004.

[27] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[28] R.M. Bolle, N.K. Ratha, and S. Pankanti. An evaluation of error confidence interval estimation methods. In *Proc. 17th Int. Conf. on Pattern Recognition (ICPR 2004)*, volume 3, pages 103–106, August 2004.

[29] (Ruud M.) Bolle, (Jonathan H.) Connell, Sharath Pankanti, (Nalini K.) Ratha, and (Andrew W.) Senior. *Guide to Biometrics*. Springer-Verlag, New-York, 2003.

[30] Kenneth A. Bollen. *Structural Equations with Latent Variables*. Wiley, 1989.

[31] Jean-François Bonastre, Frédéric Wils, and Sylvain Meignier. ALIZE, a free toolkit for speaker recognition. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 737–740, Philadelphia, USA, March 2005.

[32] Christian Borgelt and Rudolf Kruse. Local structure learning in graphical models. In *Proc. 6th ISSEK Int. Workshop on Planning Based on Decision Theory*, pages 99–118, Udine, Italy, 2002.

[33] Christian Borgelt and Rudolf Kruse. *Graphical models: methods for data analysis and mining*. John Wiley and Sons, 2002.

[34] F. Botti, A. Alexander, and A. Drygajlo. On compensation of mismatched recording conditions in the bayesian approach for forensic automatic speaker recognition. *Forensic Science International*, 146(S1):S101–S106, December 2004.

[35] Craig Boutilier, Nir Friedman, Moises Goldszmidt, and Daphne Koller. Context-specific independence in bayesian networks. In *Proc. 12th Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages 115–12, San Francisco, CA, 1996. Morgan Kaufmann.

[36] J.-J. Brault and R. Plamondon. How to detect problematic signers for automatic signature verification. In *Proc. Int. Carnahan Conf. on Security Technology*, pages 127–132, Oct. 3-5, 1989.

[37] J.-J. Brault and R. Plamondon. Segmenting handwritten signatures at their perceptually important points. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(9):953–957, Sept. 1993.

[38] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996.

[39] Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. *Classification and Regression Trees*. CRC Press, 1984.

[40] S. Bridle and England M. D. Brown. An experimental automatic word recognition system. JSRU Report 1003, Joint Speech Research Unit, Ruislip, 1974.

[41] William J. Burns and Robert T. Clemen. Covariance structure models and influence diagrams. *Management Science*, 39(7):816–834, 1993.

[42] M. P. Caligiuri, H. L. Teulings, J. V. Filoteo, D. Song, and J. B. Lohr. Quantitative measurement of handwriting in the assessment of drug-induced parkinsonism. In *Proc. 12th Conf. of the International Graphonomics Society (IGS2005)*, 2005.

[43] J.L Camino, C.M. Travieso, C.R. Morales, and M.A. Ferrer. Signature classification by Hidden Markov Model. In *Proc. IEEE 33rd Annual 1999 International Carnahan Conference on Security Technology*, pages 481–484, Oct. 1999.

[44] J.P. Jr. Campbell and D.A. Reynolds. Corpora for the evaluation of speaker recognition systems. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'99)*, volume 2, pages 829–832, March 1999.

[45] W.M. Campbell, D.A. Reynolds, J.P. Campbell, and K.J. Brady. Estimating and evaluating confidence for forensic speaker recognition. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 717–720, 2005.

[46] W.M. Campbell, D.E. Sturim, D.A. Reynolds, and A. Solomonoff. Svm based speaker verification using a gmm supervector kernel and nap variability compensation. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, 2006.

[47] D. Chan, A. Fourcin, B. Gibbon, D.and Granstrom, M. Huckvale, G. Kokkinakis, K. Kvale, L. Lamel, B. Lindberg, A. Moreno, J. Mouropoulos, F. Senia, I. Trancoso, C. Veld, and J. Zeiliger. EUROM - a spoken language resource for the EU. In *Proceedings 4th European Conference on Speech Communication and Speech Technology (Eurospeech'95)*, volume 1, pages 867–870, Madrid, Spain, September 1995.

[48] H.-D. Chang, J.-F. Wang, and H.-M. Suen. Dynamic handwritten chinese signature verification. In *Proc. second IEEE International Conference on Document Analysis and Recognition*, pages 258–261, 1993.

[49] Zhengang Chen and Xiaoqing Ding. Rejection algorithm for mis-segmented characters in multilingual document recognition. In *Proc. Int. Conf. on Document Analysis and Recognition (ICDAR)*, pages 746–749, 2003.

[50] Jie Cheng, David A. Bell, and Weiru Liu. An algorithm for bayesian belief network construction from data. In *Proc. 6th International Workshop on Artificial Intelligence and Statistics*, Fort Lauderdale, USA, 1997.

[51] David Maxwell Chickering. Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498, 2002.

[52] David Maxwell Chickering and Christopher Meek. On the incompatibility of faithfulness and monotone dag faithfulness. *Artificial Intelligence*, 170(8-9):653–666, June 2006.

[53] Samuel Chindaro, Konstantinos Sirlantzis, and Michael Fairhurst. Modelling multiple-classifier relationships using bayesian belief networks. In *Proc. 7th Int. Workshop on Multiple Classifier Systems*, pages 312–321, 2007.

[54] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Information Theory*, 14(3):462–467, 1968. ISSN 0018-9448.

[55] A.S. Constantinidis, M.C. Farihurst, and A.F.R. Rahman. A new multi-expert decision combination algorithm and its application to the detection of circumscribed masses in digital mammograms. *Pattern Recognition*, 34(8):1527–1537, 2001.

[56] Gregory F. Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, October 1992. doi: 10.1023/A:1022649401552.

[57] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. Reliability parameters to improve combination strategies in multi-expert systems. *Pattern Analysis and Applications*, 2(3):205–214, 1999.

[58] Nicandro Cruz-Ramirez, Hector-Gabriel Acosta-Mesa, Rocio-Erandi Barrientos-Martinez, and Luis-Alonso Nava-Fernandez. How good are the bayesian information criterion and the minimum description length principle for model selection? a bayesian network analysis. In *Proc. 5th Mexican Int. Conf. on Artificial Intelligence*, pages 494–504, 2006.

[59] Donald P. D'Amato. Best practices for taking face photographs and face image quality metrics. NIST Biometric Quality Workshop, March 2006.

[60] Sarat C. Dass, Yongfang Zhu, and Anil K. Jain. Validating a biometric authentication system: Sample size requirements. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2006. (to appear).

[61] J. Davis, V. Santos Costa, I. Ong, D. Page, , and I. Dutra. Using bayesian classifiers to combine rules. In *Proc. 3rd Workshop on Multi-Relational Data Mining (MRDM)*, August 2004.

[62] A. P. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(1):1–31, 1979.

[63] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(2):142–150, February 1989.

[64] Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113(1-2):41–85, September 1999.

[65] N. Dehak and G. Chollet. Support vector gmms for speaker verification. In *Proc. IEEE Odyssey 2006: The Speaker and Language Recognition Workshop*, pages 1–4, 2006.

[66] A. P. Dempster. Covariance selection. *Biometrics*, 28:157–175, 1972.

[67] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Serie B*, 39(1):1–38, 1977.

[68] Damien Dessimoz, Jonas Richiardi, Christophe Champod, and Andrzej Drygajlo. Multimodal biometrics for identity documents: State-of-the-art. Technical Report PFS 341-08.05, University of Lausanne and EPFL, September 2005.

[69] Damien Dessimoz, Jonas Richiardi, Christophe Champod, and Andrzej Drygajlo. Multimodal biometrics for identity documents (MBioID). *Forensic Science International*, 167:154–159, April 2007. doi: 10.1016/j.forsciint.2006.06.037.

[70] G. Dimauro, S. Impedovo, R. Modugno, G. Pirlo, and L. Sarcinella. Analysis of stability in hand-written dynamic signatures. In *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pages 259–263, 6-8 Aug. 2002. doi: 10.1109/IWFHR.2002.1030919.

[71] George Doddington, Walter Liggett, Alvin Martin, Mark Przybocki, and Douglas A. Reynolds. SHEEP, GOATS, LAMBS and WOLVES: a statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation. In *Proc. 5th Int. Conf. on Spoken Language Processing (ICSLP)*, Sydney, Australia, November-December 1998.

[72] G.R. Doddington. *A method of speaker verification*. Ph.D thesis, University of Wisconsin, Madison, USA, 1970.

[73] J.G.A. Dolfing, E.H.L Aarts, and J.J.G.M. van Oosterhout. On-line signature verification with Hidden Markov Models. In *Proc. International Conference on Pattern Recognition 1998*, volume 2, pages 1309–1312, Aug. 1998.

[74] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, 2nd edition, 2001.

[75] Robert P. W. Duin and David M. J. Tax. Classifier conditional posterior probabilities. In *Proc. Joint IAPR Int. Workshops SSPR '98 and SPR '98*, volume 1451 of *Lecture Notes in Computer Science*, pages 611–619, Sydney, Australia, August 1998. Springer. doi: 10.1007/BFb0033222.

[76] Bruno Dumas, Jean Hennebert, Andreas Humm, Rolf Ingold, Dijana Petrovska, Catherine Pugin, and Didier Von Rotz. Myidea - sensors specifications and acquisition protocol. DIUF-RR 2005.01, University of Fribourg, 2005.

[77] Thiago Dutra, Anne M. P. Canuto, and Marcilio C. P. de Souto. Using weighted combination-based methods in ensembles with different levels of diversity. In *Proc. 13th Int. Conf. on Neural Information Processing (ICONIP 2006)*, pages 708–717, Hong Kong, China, October 2006.

[78] Mounir El-Maliki. *Speaker verification with missing features in noisy environments*. PhD thesis, Swiss Federal Institute of Technology Lausanne (EPFL), 2000.

[79] Engin Erzin, Y. Yemez, and A.M. Tekalp. Multimodal speaker identification using an adaptive classifier cascade based on modality reliability. *Multimedia, IEEE Transactions on*, 7(5):840–852, 2005. ISSN 1520-9210. doi: 10.1109/TMM.2005.854464.

[80] K.R. Farrell. Adaptation of data fusion-based speaker verification models. In *Proceedings IEEE International Symposium on Circuits and Systems (ISCAS'02)*, volume 2, pages 851–854, 2002.

[81] Marcos Faundez-Zanuy. On-line signature recognition based on VQ-DTW. *Pattern Recognition*, In Press, Corrected Proof:–, 2006.

[82] Julian Fierrez, Javier Ortega-Garcia, Daniel Ramos, and Joaquin Gonzalez-Rodriguez. Hmm-based on-line signature verification: Feature extraction and signature modeling. *Pattern Recognition Letters*, 28(16):2325–2334, December 2007.

[83] J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez. Target dependent score normalization techniques and their application to signature verification. *IEEE Trans. on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 35(3):418–425, 2005. ISSN 1094-6977.

[84] Julian Fierrez-Aguilar, Javier Ortega-Garcia, Joaquin Gonzalez-Rodriguez, and Josef Bigun. Discriminative multimodal biometric authentication based on quality measures. *Pattern Recognition*, 38(5):777–779, May 2005.

[85] Julian Fierrez-Aguilar, Yi Chen, Javier Ortega-Garcia, and Anil K. Jain. Incorporating image quality in multi-algorithm fingerprint verification. In *Proc. Int. Conf. on Biometrics*, volume 3832 of *LNCS*, pages 213–220, Hong Kong, January 2006. Springer.

[86] Pasquale Foggia, Carlo Sansone, Gennaro Percannella, and Mario Vento. Evaluating classification reliability for combining classifiers. In *Proc. IAPR Int. Conf. on Image Analysis and Processing ICIAP*, 2007.

[87] C. Fredouille, J.-F. Bonastre, and T. Merlin. AMIRAL: A block-segmental multirecognizer architecture for automatic speaker recognition. *Digital Signal Processing*, 10(1):172–197, 2000.

[88] S.E. Fredrickson and L. Tarassenko. Text-independent speaker recognition using neural network techniques. In *Proceedings Fourth International Conference on Artificial Neural Networks*, pages 13–18, June 1995.

[89] D.K. Freeman, G. Cosier, C.B. Southcott, and I. Boyd. The voice activity detector for the pan-european digital cellular mobile telephone service. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 369–372, 1989.

[90] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, September 1995.

[91] B.J. Frey and N. Jojic. A comparison of algorithms for inference and learning in probabilistic graphical models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(9): 1392–1416, 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.169.

[92] Nir Friedman and Moises Goldszmidt. *Learning in graphical models*, chapter Learning Bayesian networks with local structure. MIT Press, 1998.

[93] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–167, 1997.

[94] D. Fuentes, D. Mostefa, J. Kharroubi, S. Garcia-Salicetti, B. Dorizzi, and G. Chollet. Identity verification by fusion of biometric data: on-line signatures and speech. In *Proc. COST 275 Workshop on the Advent of Biometrics on the Internet*, pages 83–86, Nov. 2002.

[95] M. Fuentes, S. Garcia-Salicetti, and B. Dorizzi. On line signature verification: fusion of a Hidden Markov Model and a neural network via a Support Vector Machine. In *Proc. International Workshop on Frontiers in Handwriting Recognition 2002*, pages 253–258, 2002.

[96] S. Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Trans.on Acoustics, Speech, and Signal Processing*, 29(2):254–272, April 1981.

[97] D. Garcia-Romero, J. Fierrez-Aguilar, Joaquin Gonzalez-Rodriguez, and Javier Ortega-Garcia. On the use of quality measures for text-independent speaker recognition. In *Proc. ODYSSEY 2004 - the Speaker and Language Recognition Workshop*, pages 105–110, Toldeo, Spain, May-June 2004.

[98] Daniel Garcia-Romero, Julian Fierrez-Aguilar, Joaquin Gonzalez-Rodriguez, and Javier Ortega-Garcia. Using quality measures for multilevel speaker recognition. *Computer Speech and Language*, 20(2-3):192–209, 2006.

[99] S. Garcia-Salicetti, C. Beumier, G. Chollet, B. Dorizzi, J. Leroux les Jardins, J. Lunter, Y. Ni, and D. Petrovska-Delacrtaz. Biomet: a multimodal person authentication database including face, voice, fingerprint, hand and signature modalities. In *Proc. 4th Int. Conf. on Audio and Video-Based Biometric Person Authentication (AVBPA)*, Guildford, UK, 2003.

[100] Sonia Garcia-Salicetti, Mohamed Anouar Mellakh, Lorene Allano, and Bernadette Dorizzi. Multimodal biometric score fusion: the mean rule vs. support vector classifiers. In *Proc. 13th European Signal Processing Conference (EUSIPCO)*, 2005.

[101] A. Garg, V. Pavlovic, and T.S. Huang. Bayesian networks as ensemble of classifiers. In *Proc. 16th Int. Conf. on Pattern Recognition (ICPR)*, volume 2, pages 779–784, 2002.

[102] John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, Nancy L. Dahlgren, and Victor Zue. Timit acoustic-phonetic continuous speech corpus, 1993. LDC catalog number LDC93S1.

[103] J.-L. Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Trans. on Speech and Audio Processing*, 2(2): 291–298, April 1994. doi: 10.1109/89.279278.

[104] Yang Ge and Wenxin Jiang. On consistency of bayesian inference with mixtures of logistic regression. *Neural Computation*, 18(1):224–243, 2006.

[105] Dan Geiger and David Heckerman. Knowledge representation and inference in similarity networks and bayesian multinets. *Artificial Intelligence*, 82(1-2):45–74, April 1996.

[106] Pierre Geurts and Louis Wehenkel. Investigation and reduction of discretization variance in decision tree induction. In *Proc. 11th European Conf. on Machine Learning (ECML)*, pages 162–170, 2000.

[107] H. Gish and M. Schmidt. Text-independent speaker identification. *IEEE Signal Processing Magazine*, 11(4):18–32, October 1994. ISSN 1053-5888. doi: 10.1109/79.317924.

[108] Sabine Glesner and Daphne Koller. Constructing flexible dynamic belief networks from first-order probabilistic knowledge bases. In *Proc. European Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 217–226, 1995.

[109] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5):533–549, 1986.

[110] J. Godfrey, D. Graff, and A. Martin. Public databases for speaker recognition and verification. In *Proceedings ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 39–42, Martigny, Switzerland, April 1994.

[111] Robert M. Gray and Lee D. Davisson. *An Introduction to Statistical Signal Processing*. Cambridge University Press, 2004.

[112] Yong Gu and Trevor Thomas. A text-independent speaker verification system using support vector machines classifier. In *Proceedings 7th European Conference on Speech Communication and Technology (EUROSPEECH 2001 Scandinavia)*, pages 1765–1768, Aalborg, Denmark, September 2001.

[113] S. Guruprasad, N. Dhananjaya, and B. Yegnanarayana. AANN models for speaker recognition based on difference cepstrals. In *Proceedings International Joint Conference on Neural Networks*, volume 1, pages 692–697, July 2003.

[114] I. Guyon, CF. Aliferis, and A. Elisseeff. *Computational Methods of Feature Selection*, chapter Causal Feature Selection. Chapman and Hall, 2007.

[115] Andrew Hamilton-Wright and Daniel W. Stashuk. A decision support framework for clinical needle emg. In *Proc. 17th IASTED Int. Conf. on Modelling and simulation (MS'06)*, pages 116–121, Anaheim, CA, USA, 2006. ACTA Press. ISBN 0-88986-592-2.

[116] Frank Harary. *Graph Theory.* Addison-Wesley, 1995.

[117] J. Harmse, S.D. Beck, and H. Nakasone. Speaker recognition score-normalization to compensate for snr and duration. In *IEEE Odyssey 2006: The Speaker and Language Recognition Workshop*, pages 1–8, June 2006. doi: 10.1109/ODYSSEY.2006.248092.

[118] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning.* Springer, 2001.

[119] L.P. Heck and M. Weintraub. Handset-dependent background models for robust text-independent speaker recognition. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 1071–1074, April 1997. doi: 10.1109/ICASSP.1997.596126.

[120] David Heckerman and Dan Geiger. Learning bayesian networks: a unification for discrete and gaussian domains. In *Proc. 11th Conf. on uncertainty in artificial intelligence*, pages 274–284, 1995.

[121] David Heckerman, Dan Geiger, and David M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

[122] H. Hermansky and N. Morgan. Rasta processing of speech. *IEEE Trans. on Speech and Audio Processing,*, 2(4):578–589, 1994. ISSN 1063-6676.

[123] Hynek Hermansky. Perceptual linear predictive (plp) analysis of speech. *The Journal of the Acoustical Society of America*, 87:1738–1752, 1990.

[124] Alan Higgins and Dave Vermilyea. King speaker verification, 1992. LDC catalog number LDC95S22.

[125] Tin Kam Ho, J.J. Hull, and S.N. Srihari. Decision combination in multiple classifier systems. *IEEE Trans. Pattern Analysis and Machine Intelligence,*, 16(1):66–75, Jan. 1994. doi: 10.1109/34.273716.

[126] Reimar Hofmann and Volker Tresp. Discovering structure in continuous variables using bayesian networks. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 500–506. The MIT Press, 1996.

[127] Harry Hollien, Gea Dejong, Camilo A. Martin, Reva Schwartz, and Kristen Liljegren. Effects of ethanol intoxication on speech suprasegmentals. *Acoustical Society of America Journal*, 110 (6):3198–3206, 2001.

[128] Xuelei Hu and Lei Xu. A comparative study of several cluster number selection criteria. In *Proc. 4th Int. Conf. on Intelligent Data Engineering and Automated Learning (IDEAL)*, pages 195–202, 2003.

[129] C. Huang and A. Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):255–263, 1996.

[130] Wei Huang and Yaxin Zhang. Online adaptive score normalization for noise robustness speaker verification on cellular phone. In *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*, pages 1–5, 2006.

[131] X. D. Huang and M. A. Jack. Semi-continuous hidden markov models for speech signals. *Computer Speech and Language*, 3(3):239–251, July 1989.

[132] Y.S. Huang and C.Y. Suen. A method of combining multiple experts for the recognition ofunconstrained handwritten numerals. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(1):90–94, 1995. ISSN 0162-8828.

[133] Mark C. Huggins and John J. Grieco. Confidence metrics for speaker identification. In *Proc. 7th Int'l Conf. on Spoken Language Processing (ICSLP)*, 2002.

[134] J. Ilonen, P. Paalanen, J.-K. Kamarainen, and H. Kalviainen. Gaussian mixture pdf in one-class classification: computing and utilizing confidence values. In *18th Int. Conf. on Pattern Recognition (ICPR)*, volume 2, pages 577–580, 2006.

[135] A.K. Jain, F.D. Griess, and S.D. Connell. On-line signature verification. *Pattern Recognition*, 35:2963–2972, 2002.

[136] (Anil K.) Jain, (Robert P. W.) Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[137] Finn V. Jensen. *Bayesian networks and decision graphs*. Springer, 2001.

[138] F.V. Jensen. *Introduction to Bayesian networks*. Springer-Verlag New York, 1996.

[139] Hui Jiang. Confidence measures for speech recognition: A survey. *Speech Communication*, 45 (4):455–470, April 2005.

[140] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Proc. 11th Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 338–345, Montreal, Canada, 1995.

[141] George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 121–129, 1994.

[142] I. Karlsson, T. Banziger, T. Dankovicová, J.and Johnstone, J. Lindberg, H. Melin, F. Nolan, and K. Scherer. Speaker verification with elicited speaking styles in the verivox project. *Speech Communication*, 31(2-3):121–129, June 2000.

[143] R. Kashi, J. Hu, W.L. Nelson, and W. Turin. A Hidden Markov Model approach to online handwritten signature recognition. *International Journal on Document Analysis and Recognition*, 1(2):102–109, 1998.

[144] Ramanujan S. Kashi, William T Turin, and Winston L. N. Nelson. On-line handwritten signature verification using stroke direction coding. *Optical Engineering*, 35(9):2526–2533, September 1996.

[145] M. Kearns, Y. Mansour, A. Ng, and D. Ron. An experimental and theoretical comparison of model selection methods. *Machine Learning*, 27:7–50, 1997.

[146] H. Ketabdar, J. Richiardi, and A. Drygajlo. Global feature selection for on-line signature verification. In *Proc. 12th Conference of the International Graphonomics Society*, pages 59–63, Salerno, Italy, June 2005.

[147] Yurij Kharin. *Robustness in statistical pattern recognition.* Kluwer Academic Publishers, 1996.

[148] Minyoung Kim and V. Pavlovic. Discriminative learning of mixture of bayesian network classifiers for sequence classification. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 268–275, 2006. doi: 10.1109/CVPR.2006.101.

[149] S. Kirkpatrick, C. D. Gelatt, and J. M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[150] J. Kittler, J. Matas, K. Jonsson, and M. Ramos Sánchez. Combining evidence in personal identity verification systems. *Pattern Recognition Letters*, 18:845–852, 1997.

[151] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998. ISSN 0162-8828. doi: 10.1109/34.667881.

[152] J. Kittler, M. Ballette, J. Czyz, F. Roli, and L. Vandendorpe. Decision level fusion of intramodal personal identity verification experts. In *Proc. Int. Workshop on Multiple Classifier Systems*, pages 1–4, 2002.

[153] J Kittler, N Poh, O Fatukasi, K Messer, K Kryszczuk, J Richiardi, and A Drygajlo. Quality dependent fusion of intramodal and multimodal biometric experts. In *Proc. SPIE Defense and Security Symposium*, Orlando, USA, April 2007.

[154] Josef Kittler, Kerion Messer, Omolara Fatukasi, Andrzej Drygajlo, Jonas Richiardi, and Krzysztof Kryszczuk. Biometric data quality and expert confidence dependent fusion of multiple modalities. Biosecure Vigo Workshop, oral presentation, 2006.

[155] Uffe Kjaerulff. Triangulation of graphs — algorithms giving small total state space. Research Report R 90-09, University of Aalborg, Dept. of Mathematics and Computer Science, Institute for Electronic Systems, March 1990.

[156] Gudrun Klasmeyer, Tom Johnstone, Tanja Bänziger, Christopher Sappok, and Klaus R. Scherer. Emotional voice variability in speaker verification. In *Proc. ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion*, pages 213–218, 2000.

[157] G. D. Kleiter and R Jirousek. Learning bayesian networks under the control of mutual information. In *Proc. Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 985–990, 1996.

[158] A.L. Koerich. Rejection strategies for handwritten word recognition. In *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*, pages 479–484, 2004. doi: 10.1109/IWFHR.2004.88.

[159] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI).*, 1995.

[160] D. Koller and U. Lerner. Sampling in factored dynamic systems. In A. Doucet, J.F.G. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods In Practice*. Springer-Verlag, 2000.

[161] Eun Bae Kong and Thomas G. Dietterich. Error-correcting output coding corrects bias and variance. In *Proc. Int. Conf. on Machine Learning (ICML)*, 1995.

[162] Johan Koolwaaij and Lou Boves. On decision making in forensic casework. *The International Journal of Speech, Language and the Law: Forensic Linguistics*, 6(2):242–164, 1999.

[163] Kevin B. Korb and Ann E. Nicholson. *Bayesian artificial intelligence*. Chapman and Hall, 2004.

[164] Krzysztof Kryszczuk and Andrzej Drygajlo. Reliability measures and error prediction in biometric identity verification. *Journal of Signal Processing*, 2006. (submitted).

[165] Krzysztof Kryszczuk, Jonas Richiardi, Plamen Prodanov, and Andrzej Drygajlo. Error handling in multimodal biometric systems using reliability measures. In *Proc. 12th European Conference on Signal Processing (EUSIPCO)*, Antalya, Turkey, September 2005.

[166] Krzysztof Kryszczuk, Jonas Richiardi, and Andrzej Drygajlo. Reliability estimation for multimodal error prediction and fusion. In *Proc. 7th Int. Workshop on Pattern Recognition in Information Systems (PRIS 2007)*, Funchal, Portugual, June 2007.

[167] Krzysztof Kryszczuk, Jonas Richiardi, Plamen Prodanov, and Andrzej Drygajlo. Reliability-based decision fusion in multimodal biometric verification systems. *EURASIP Journal of Advances in Signal Processing*, 2007, 2007. doi: 10.1155/2007/86572.

[168] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *Proc. Int'l Conf. on Machine Learning (ICML)*, pages 179–186, 1997.

[169] Matjaz Kukar and Ciril Groselj. Transductive machine learning for reliable medical diagnostics. *Journal of Medical Systems*, V29(1):13–32, February 2005.

[170] Ludmila I. Kuncheva. On the optimality of naive bayes with dependent binary features. *Pattern Recognition Letters*, 27(7):830–837, May 2006.

[171] Ludmila Ilieva Kuncheva. *Combining Pattern Classifiers*. Wiley and sons, 2004.

[172] Amlan Kundu, Yang He, and Paramvir Bahl. Recognition of handwritten word: First and second order hidden markov model based approach. *Pattern Recognition*, 22(3):283–297, 1989.

[173] Xiangyang Lan and D.P. Huttenlocher. Beyond trees: common-factor models for 2d human pose recovery. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 470–477, 2005.

[174] Pat Langley and Stephanie Sage. Induction of selective bayesian classifiers. In *Proc. 10th Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, San Francisco, CA, 1994. Morgan Kaufmann.

[175] Pat Langley, Wayne Iba, and Kevin Thompson. An analysis of bayesian classifiers. In *Proc. 10th National Conf. on Artificial Intelligence*, San Jose, USA, July 1992. AAAI Press.

[176] P. Larranaga, C.M.H. Kuijpers, R.H. Murga, and Y. Yurramendi. Learning bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE Trans. on Systems, Man and Cybernetics, Part A*, 26(4):487–493, 1996. ISSN 1083-4427.

[177] K. K. Lau, P. C. Yuen, and Y. Y. Tang. *Advances in Handwriting Recognition*, chapter An efficient Function-based On-line Signature Recognition System, pages 559–568. World Scientific, 1999.

[178] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50(2):157–224, 1988.

[179] Steffen Fl Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[180] J. A. Leonard, M. A. Kramer, and L. H. Ungar. A neural network architecture that computes its own reliability. *Computers and Chemical Engineering*, 16(9):819–835, September 1992.

[181] P. Leray, H. Zaragoza, and F. d'Alché-Buc. Pertinence des mesures de confiance en classification. In *12ème Congrès Francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Articifielle (RFIA 2000)*, pages 267–276, Paris, France, 2000.

[182] Ying Liu, Martin Russell, and Michael Carey. Speaker recognition using a trajectory-based segmental HMM. In *Proc. ODYSSEY 2004 - The Speaker and Language Recognition Workshop*, pages 45–50, 2004.

[183] Marcus Liwicki, Andreas Schlapbach, Horst Bunke, Samy Bengio, Johnny Mariéthoz, and Jonas Richiardi. Writer identification for smart meeting room systems. In *Proc. 7th IAPR International Workshop on Document Analysis Systems*, volume 3872 of *Lecture Notes in Computer Science*, pages 186–195, Nelson, New Zealand, February 2006. doi: DOI:10.1007/11669487_17.

[184] Jérôme Louradour, Khalid Daoudi, and Francis Bach. Svm speaker verification using an incomplete cholesky decomposition sequence kernel. In *Proc. IEEE Odyssey 2006: The Speaker and Language Recognition Workshop*, 2006.

[185] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2004.

[186] D. Madigan, J. Gavrin, and A. Raftery. Eliciting prior information to enhance the predictive performance of bayesian graphical models. *Communications in statistics. Theory and methods*, 24(9):2271–2292, 1995.

[187] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assessment of decision task performance. In *Proc. Eurospeech 1997*, pages 1895–1898, 1997.

[188] Ofer Matan. On voting ensembles of classifiers (extended abstract). In *Working Notes of the Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, Portland, USA, 1996. held in conjunction with the 13th Nat. Conf. on Artificial Intelligence (AAAI-96).

[189] T. Matsui and S. Furui. Comparison of text-independent speaker recognition methods using vq-distortion and discrete/continuous hmm's. *Speech and Audio Processing, IEEE Transactions on*, 2(3):456–459, July 1994. ISSN 1063-6676.

[190] Geoffrey J. McLachlan and Kaye E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, 1987.

[191] H. Melin. Databases for speaker recognition: Activities in COST250 working group 2. In *Proceedings COST250 Workshop on Speaker Recognition in Telephony*, Rome, Italy, Nov 1999.

[192] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: The extended M2VTS database. In *Proc. 2nd Int. Conf. on Audio- and Video-Based Biometric Person Authentication (AVBPA)*, pages 72–77, 1999.

[193] W.B. Mikhael and P. Premakanthan. Speaker recognition employing waveform based signal representation in nonorthogonal multiple transform domains. In *Proceedings IEEE International Symposium on Circuits and Systems (ISCAS'02)*, volume 2, pages 608–611, May 2002.

[194] B. Miller. Vital signs of identity. *IEEE Spectrum*, 31(2):22–30, 1994.

[195] Ji Mingy, Timothy J. Hazenz, and James R. Glassz. A comparative study of methods for handheld speaker verification in realistic noisy conditions. In *Proc. IEEE Odyssey 2006: The Speaker and Language Recognition Workshop*, 2006.

[196] G. Monaci, P. Jost, P. Vandergheynst, B. Mailhe, S. Lesage, and R. Gribonval. Learning Multi-Modal Dictionaries. *IEEE Transactions on Image Processing*, 16(9):2272–2283, 2007. doi: NA.

[197] D. Muramatsu and T. Matsumoto. An HMM on-line signature verification algorithm. In *Proc. International Conferencence on Audio- and Video-Based Biometric Person Authentication 2003*, pages 233–241, Jun. 2003.

[198] K. Murphy. *Dynamic Bayesian networks: representation, inference and learning*. PhD thesis, University of California at Berkeley, July 2002.

[199] Kevin Murphy. Inference and learning in hybrid bayesian networks. Technical Report CSD-98-990, U.C. Berkeley, Computer Science Division, January 1998.

[200] I. Nakanishi, H. Sakamoto, Y. Itoh, and Y. Fukui. Multi-matcher on-line signature verification system in DWT domain. In *Proc. 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 965–968, Philadelphia, USA, March 2005.

[201] Hirotaka Nakasone and Steven D. Beck. Forensic automatic speaker recognition. In *Proc. 2001: A Speaker Odyssey*, 2001.

[202] V.S. Nalwa. Automatic on-line signature verification. *Proc. of the IEEE*, 82(2):215–239, February 1997.

[203] K. Nandakumar, Yi Chen, A.K. Jain, and S.C. Dass. Quality-based score level fusion in multibiometric systems. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 473–476, 2006.

[204] National Institute of Standards and Technology. The 2001 NIST evaluation plan for recognition of conversational speech over the telephone, Oct. 2000.

[205] Radford M. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 56(1): 71–113, July 1992.

[206] Richard E. Neapolitan. Computing the confidence in a medical decision obtained from an influence diagram. *Artificial Intelligence in Medicine*, 5(4):341–363, August 1993.

[207] E. Nemer, R. Goubran, and S. Mahmoud. Snr estimation of speech signals using subbands and fourth-order statistics. *IEEE Signal Processing Letters*, 6(7):171–174, 1999. ISSN 1070-9908.

[208] N.B. Nill and B.H. Bouzas. Objective image quality measure derived from digital image power spectra. *Optical Engineering*, 31(4):813–825, April 1992. doi: 10.1117/12.56114.

[209] NIST Smart Space Project. NIST speech signal to noise ratio measurements. http://www.nist.gov/smartspace/tools.html.

[210] J.S. Oglesby, J. Mason. Speaker recognition with a neural classifier. In *Speech 88: Proceedings of the 7th Federation of Acoustical. Societies of Europe (FASE) Symposium*, pages 1357–1363, Edinburgh, UK, 1988.

[211] J.S. Oglesby, J. Mason. Speaker recognition with a neural classifier. In *Proceedings First IEE International Conference on artificial Neural Networks*, volume 313, pages 306–309, October 1989.

[212] J.S. Oglesby, J. Mason. Radial basis function networks for speaker recognition. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'91)*, volume 1, pages 393–396, April 1991.

[213] J.P. Openshaw and J.S. Mason. On the limitations of cepstral features in noise. In *Proc. IEEE ICASSP*, pages 49–52, April 1994.

[214] Julio Ortega, Moshe Koppel, and Shlomo Argamon. Arbitrating among competing classifiers using learned referees. *Knowledge and Information Systems*, V3(4):470–490, November 2001.

[215] Javier Ortega-García and Joaquín González-Rodríguez. Overview of speech enhancement techniques for automatic speaker recognition. In *4th Int. Conf. on Spoken Language Processing ICSLP*, pages 929–932, 1996.

[216] J. Ortega-Garcia, J. Gonzalez-Rodriguez, A. Simon-Zorita, and S. Cruz-Llanas. From biometrics technology to applications regarding face, voice, signature and fingerprint recognition systems. In D.D. Zhang, editor, *Biometric solutions for authentication in a e-world*. Kluwer Academic Publishers, July 2002.

[217] J. Ortega-Garcia, J. Fierrez-Aguilar, J. Martin-Rello, and J. Gonzalez-Rodriguez. Complete signal modeling and score normalization for function-based dynamic signature verification. In *Proc. International Conference on Audio- and Video-Based Biometric Person Authentication 2003*, pages 658–667, Guildford, UK, 2003.

[218] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, (J.-J.) Igarza, C. Vivaracho, D. Escudero, and (Q.-I.) Moro. MCYT baseline corpus: A bimodal biometric database. In *IEE Proceedings - Vision, Image and Signal Processing*, volume 150, pages 395–401, 2003.

[219] Javier Ortega-Garcia, Joaquin Gonzalez-Rodriguez, and Victoria Marrero-Aguiar. AHUMADA: A large speech corpus in spanish for speaker characterization and identification. *Speech Communication*, 31:255–264, 2000.

[220] Astrid Paeschke and Walter F. Sendlmeier. Prosodic characteristics of emotional speech: Measurements of fundamental frequency movements. In *Proc. ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion*, pages 75–80, September 2000.

[221] M. Pandit and J. Kittler. Feature selection for a DTW-based speaker verification system. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'98)*, volume 2, pages 769–772, 1998.

[222] M. Parizeau and R. Plamondon. A comparative analysis of regional correlation, dynamic time warping, and skeletal tree matching for signature verification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(7):710–717, July 1990.

[223] E.K. Patterson, S. Gurbuz, Z. Tufekci, and J.N. Gowdy. Moving-talker, speaker-independent feature study, and baseline results using the cuave multimodal speech corpus. *EURASIP Journal on Applied Signal Processing*, 2002(11):1189–201, November 2002. ISSN 1110-8657.

[224] Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288, September 1986.

[225] Judea Pearl. *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufmann, 1988.

[226] J. Pelecanos, J. Navratil, and G.N. Ramaswamy. Addressing channel mismatch through speaker discriminative transforms. In *Proc. IEEE Odyssey 2006: The Speaker and Language Recognition Workshop*, pages 1–6, 2006.

[227] Jason Pelecanos and Sridha Sridharan. Feature warping for robust speaker verification. In *Proc. 2001: A Speaker Odyssey - The Speaker Recognition Workshop*, pages 213–218, 2001.

[228] Stephane Pigeon, Pascal Druyts, and Patrick Verlinde. Applying logistic regression to the fusion of the nist'99 1-speaker submissions. *Digital Signal Processing*, 10(1-3):237–248, January 2000.

[229] John Pitrelli and Michael Perrone. Confidence modeling for verification post-processing for handwriting recognition. In *Proc. Eighth Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pages 30–35, Niagara-on-the-Lake, Canada, August 2002. doi: 10.1109/IWFHR.2002.1030880.

[230] R. Plamondon and G. Lorette. On-line signature verification: how many countries are in the race? In *Proc. IEEE International Carnahan Conference on Security Technology*, pages 183–191, Zuerich, Switzerland, 1989.

[231] J. Platt. *Advances in Kernel Methods – Support Vector Learning*, chapter Fast Training of Support Vector Machines using Sequential Minimal Optimization. MIT Press, 1998.

[232] Norman Poh and Samy Bengio. Improving fusion with margin-derived confidence in biometric authentication tasks. In *Fifth Int. Conf. Audio- and Video-Based Biometric Person Authentication (AVBPA)*, 2005.

[233] Norman Poh and Samy Bengio. Database, protocols and tools for evaluating score-level fusion algorithms in biometric authentication. *Pattern Recognition*, 39(2):223–233, February 2006.

[234] Norman Poh, Guillaume Heusch, and Josef Kittler. On combination of face authentication experts by a mixture of quality dependent fusion classifiers. In *Proc. 7th Int. Workshop on Multiple Classifier Systems*, pages 344–356, Prague, Czech Republic, 2007.

[235] A. Poritz. Linear predictive hidden markov models and the speech signal. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'82)*, volume 7, pages 1291–1294, May 1982.

[236] P. Prodanov, J. Richiardi, and A. Drygajlo. Graphical models for dialogue repair in multimodal interaction with service robots. In *Proc. 8th COST276 workshop*, Trondheim, Norway, May 2005.

[237] Plamen Prodanov, Andrzej Drygajlo, Jonas Richiardi, and Anil Alexander. Grounding in multimodal service robot conversational system using graphical models. *Journal of Intelligent Service Robotics*, 2007. (In press).

[238] P. Pudil, J. Novovicova, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125, November 1994.

[239] Joanna Putz-Leszczynska and Andrzej Pacut. Dynamic time warping in subspaces for on-line signature verification. In *Proc. 12th Biennial Conference of the International Graphonomics Society*, pages 108–112, Salerno, Italy, June 2005.

[240] J. R. Quinlan. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.

[241] J.R. Quinlan. Induction of decision trees. *Machine Learning*, V1(1):81–106, March 1986. doi: 10.1023/A:1022643204877.

[242] Sarunas Raudys and Fabio Roli. The behavior knowledge space fusion method: Analysis of generalization error and strategies for performance improvement. In *Proc. Int. Workshop on Multiple Classifier Systems*, pages 160–160, 2003.

[243] Sarunas Raudys, Ray Somorjai, and Richard Baumgartner. Reducing the overconfidence of base classifiers when combining their decisions. In *Proc. 4th Int. Workshop on Multiple Classifier Systems (MCS)*, pages 161–161, 2003.

[244] Philippe Renevey and Andrzej Drygajlo. Entropy based voice activity detection in very noisy conditions. In *Proc. 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, 2001.

[245] D. Reynolds. *A Gaussian mixture modeling approach to text-independent speaker identification.* PhD thesis, Georgia Institute of Technology, Atlanta, USA, 1992.

[246] D.A. Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech Communication*, 17:91–108, 1995.

[247] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1–3):19–41, 2000.

[248] Douglas A. Reynolds. Comparison of background normalization methods for text-independent speaker verification. In *Proc. 5th European Conf. on Speech Communication and Technology (EUROSPEECJ)*, pages 963–966, 1997.

[249] T.H. Rhee, S.J. Cho, and J.H. Kim. On-line signature verification using model-guided segmentation and discriminative feature selection for skilled forgeries. In *Proc. Sixth International Conference on Document Analysis and Recognition*, pages 645–649, September 2001.

[250] J. Richiardi. Resilience of on-line signature verification to packet loss on IP networks: preliminary experiments. COST275 STSM Report, Fondazione Ugo Bodoni, Italy, Sept. 2003.

[251] J. Richiardi and A. Drygajlo. Gaussian mixture models for on-line signature verification. In *Proc. ACM SIGMM Multimedia, Workshop on Biometrics methods and applications (WBMA)*, pages 115–122, Berkeley, USA, Nov. 2003.

[252] J. Richiardi, J. Fierrez-Aguilar, J. Ortega-Garcia, and A. Drygajlo. On-line signature verification resilience to packet loss in IP networks. In *Proc. 2nd COST 275 Workshop on Biometrics on the Internet: fundamentals, advances and applications*, pages 9–14, Vigo, Spain, March 2004.

[253] J. Richiardi, A. Drygajlo, A. Palacios-Venin, R. Ludvig, O. Genton, and L. Houmgny. A distributed multimodal biometric authentication framework. In *Proc. 3nd COST 275 Workshop on Biometrics on the Internet*, pages 85–88, Hatfield, U.K., October 2005.

[254] J. Richiardi, P. Prodanov, and A. Drygajlo. A probabilistic measure of modality reliability in speaker verification. In *Proc. IEEE International Conf. on Acoustics, Speech and Signal Processing 2005*, pages 709–712, Philadelphia, USA, March 2005.

[255] Jonas Richiardi and Andrzej Drygajlo. Applying biometrics to identity documents: Estimating and coping with errors. SNSF project technical report, Swiss Federal Institute of Technology, October 2006.

[256] Jonas Richiardi and Andrzej Drygajlo. Applying biometrics to identity documents: Implementation issues. SNSF project technical report, Swiss Federal Institute of Technology, 2006.

[257] Jonas Richiardi and Andrzej Drygajlo. Reliability-based voting schemes using modality-independent features in multi-classifier biometric authentication. In *Proc. 7th Int. Workshop on Multiple Classifier Systems*, Prague, Czech Republic, May 2007. Springer.

[258] Jonas Richiardi and Andrzej Drygajlo. Evaluation of speech quality measures for the purpose of speaker verification. In *Proc. Odyssey 2008: The Speaker and Language Recognition Workshop*, Stellenbosch, South Africa, January 2008.

[259] Jonas Richiardi, Hamed Ketabdar, and Andrzej Drygajlo. Local and global feature selection for on-line signature verification. In *Proc. IAPR 8th International Conference on Document Analysis and Recognition (ICDAR 2005)*, volume 2, pages 625–629, Seoul, Korea, August-September 2005. doi: 10.1109/ICDAR.2005.152.

[260] Jonas Richiardi, Andrzej Drygajlo, and Plamen Prodanov. Confidence and reliability measures in speaker verification. *Journal of the Franklin Institute*, 343(6):574–595, September 2006. doi: 10.1016/j.jfranklin.2006.07.002.

[261] Jonas Richiardi, Plamen Prodanov, and Andrzej Drygajlo. Speaker verification with confidence and reliability measures. In *Proc. 2006 IEEE International Conference on Speech, Acoustics and Signal Processing*, Toulouse, France, May 2006.

[262] Jonas Richiardi, Krzysztof Kryszczuk, and Andrzej Drygajlo. Quality measures in unimodal and multimodal biometric verification. In *Proc. 15th European Signal Processing Conf. (EUSIPCO)*, Poznan, Poland, 2007.

[263] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, September 1978.

[264] J. Rissanen. *Stochastic complexity in statistical inquiry*, volume 15 of *World Scientific series in computer science*. World Scientific, Singapore, 1989.

[265] S.J. Roberts, D. Husmeier, I. Rezek, and W. Penny. Bayesian approaches to Gaussian mixture modeling. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11):1133–1142, Nov. 1998.

[266] Fabio Roli, Josef Kittler, Giorgio Fumera, and Daniele Muntoni. An experimental comparison of classifier fusion rules for multimodal personal identity verification systems. In *Proc. of the Third International Workshop on Multiple Classifier Systems*, pages 325–335, 2002.

[267] Richard Rose. Environmental robustness in automatic speech recognition. In *Proc. COST278 and ISCA Tutorial and Research Workshop on Robustness Issues in Conversational Interaction*, Norwich, UK, 2004.

[268] Aaron E. Rosenberg, Joel DeLong, Chin-Hui Lee, Biing-Hwang Juang, and Frank K. Soong. The use of cohort normalized scores for speaker verification. In *Proc. 2nd Int. Conf. on Spoken Language Processing (ICSLP)*, pages 599–602, 1992.

[269] A.E. Rosenberg, C.-H. Lee, and S.; Gokcen. Connected word talker verification using whole word hidden markov models. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'91)*, volume 1, pages 381–384, April 1991.

[270] Arun Ross and Anil Jain. Information fusion in biometrics. *Pattern Recognition Letters*, 24: 2115–2125, 2003.

[271] Arun Ross and Anil K. Jain. Multimodal biometrics: An overview. In *Proc. 12th European Signal Processing Conf.*, pages 1221–1224, 2004.

[272] Mohammad T. Sadeghi and Josef Kittler. Confidence based gating of multiple face authentication experts. In *Proc. Joint IAPR Int. Workshops on Structural, Syntactic, and Statistical Pattern Recognition (SSPR 2006, SPR 2006)*, pages 667–676, Hong Kong, China, August 2006.

[273] D. Sakamoto, H. Morita, T. Ohishi, Y. Komiya, and T. Matsumoto. On-line signature verification incorporating pen position, pen pressure and pen inclination trajectories. In *Proc. 2001 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7–11, May 2001. Dynamic Time Warping.

[274] E. Sanchez-Soto, R. Blouet, G. Chollet, and M. Sigelle. Speaker verification with bayesian networks. In *Proc. Workshop on Multimodal User Authentification (MMUA)*, Santa Barbara, USA, 2003.

[275] E. Sanchez-Soto, R. Blouet, M. Sigelle, and G. Chollet. Model adaptation for speaker verification using conditional probability tables in bayesian networks. In *Proc. COST275 Workshop on Biometrics on the Internet* ., Vigo, Spain, 2004.

[276] Conrad Sanderson and Kuldip K. Paliwal. Noise compensation in a person verification system using face and multiple speech features. *Pattern Recognition*, 36(2):293–302, February 2003.

[277] Lifeng Sang, Zhaohui Wu, and Yingchun Yang. Speaker recognition system in multi-channel environment. In *IEEE Int. Conf. on Systems, Man and Cybernetics*, volume 4, pages 3116–3121 vol.4, 2003.

[278] Y. Sato and K. Kogure. Online signature verification based on shape, motion, and writing pressure. In *Proc. 6th Int'l Conf. on Pattern Recognition*, pages 823–826, 1982.

[279] Cullen Schaffer. Selecting a classification method by cross-validation. *Machine Learning*, 13 (1):135–143, October 1993.

[280] Sascha Schimke, Athanasios Valsamakis, Claus Vielhauer, and Yannis Stylianou. Biometrics: Different approaches for using gaussian mixture models in handwriting. In *Proc. Communications and Multimedia Security (CMS 2005)*, volume 3677/2005 of *Lecture Notes in Computer Science*, pages 261–263, 2005. doi: 10.1007/11552055_26.

[281] Natalia Schmid, Nathan Kalka, Jinyu Zuo, and Bojan Cukic. Performance analysis of individual and combined quality effects for iris biometrics. In *Proc. NIST Biometric Quality Workshop*, 2006.

[282] M. Schmidt and H. Gish. Speaker identification via support vector classifiers. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'96)*, volume 1, pages 105–108, May 1996.

[283] R. Schwartz, S. Roucos, and M. Berouti. The application of probability density estimation to text-idenpendent speaker identification. In *Proceedings IEEE International Conference on Speech, Acoustics, and Signal Processing*, pages 1649–1652, 1982.

[284] Ross Shachter. Bayes-ball: The rational pasttime (for determining irrelevance and requisite information in belief networks and influence diagrams). In *Proc. 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 480–489, San Francisco, USA, 1998. Morgan Kaufmann.

[285] Ross D. Shachter and C. Robert Kenley. Gaussian influence diagrams. *Management Science*, 35(5):527–550, May 1989.

[286] David J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC press, 2004.

[287] Catherine A. Shipp and Ludmila I. Kuncheva. Relationships between combination methods and measures of diversity in combining classifiers. *Information Fusion*, 3(2):135–148, June 2002.

[288] John Simpson, editor. *Oxford English Dictionary*. Oxford Edition, 2004.

[289] Steven Skiena. *Implementing Discrete Mathematics: Combinatorics and Graph Theory With Mathematica*. Perseus Books, 1990.

[290] Yosef A. Solewicz and Moshe Koppel. Enhanced fusion methods for speaker verification. In *Proc. 9th Conf. Speech and Computer (SPECOM)*, pages 388–392, St.-Petersburg, Russia, September 2004.

[291] F.K. Soong, A.E. Rosemberg, L.R. Rabiner, and B.H. Juang. A vector quantization approach to speaker recognition. In *Proceedings IEEE International Conference on Speech, Acoustics, and Signal Processing*, pages 387–390, 1985.

[292] David J. Spiegelhalter and Steffen L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20(5):579–605, 1990. doi: 10.1002/net.3230200507.

[293] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, 2001.

[294] Speech Technology Center (St.-Petersburg). STC russian speech database, 1998. ELDA catalogue number S0050.

[295] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J.M. Buhmann. Topology free Hidden Markov Models: Application to background modeling. In *Proc. 8th International Conference on Computer Vision*, pages 294–301, 2001.

[296] A. Stolcke and S.M. Omohundro. Best-first model merging for Hidden Markov Model induction. Technical Report TR-94-003, International Computer Science Institute, Berkeley, April 1994.

[297] Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. of Machine Learning Research*, 3:619–620, 2002.

[298] Yannis Stylianou, Yannis Pantazis, Felipe Calderero, Pedro Larroy, Francois Severin, Sascha Schimke, Rolando Bonal, Federico Matta, and Athanasios Valsamakis. GMM-based multimodal biometric verification. In *Proc. eNTERFACE 2005 Summer Workshop on Multimodal Interfaces*, pages 44–51, Mons, Belgium, July-August 2005. Presses universitaires de Louvain.

[299] John A. Swets, editor. *Signal Detection and Recognition by Human Observers*, pages 611–648. Wiley, 1964.

[300] Elham Tabassi, Charles L. Wilson, and Craig I. Watson. Fingerprint image quality. NISTIR 7151, National Institute of Standards and Technology, August 2004.

[301] Remco Teunen, Ben Shahshahani, and Larry Heck. A model-based transformational approach to robust speaker recognition", in icslp-2000, vol.2, 495-498. In *Proc. 6th Int. Conf. on Spoken Language Processing (ICSLP)*, 2000.

[302] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition*. Academic Press, 2006.

[303] Clifford S. Thomas, Catherine A. Howie, and Leslie S. Smith A1. A new singly connected network classifier based on mutual information. *Intelligent Data Analysis*, 9:189–205, 2005.

[304] Kar-Ann Toh, Wei-Yun Yau, Eyung Lim, Lawrence Chen, and Chin-Hon Ng. *Fusion of Auxiliary Information for Multi-modal Biometrics Authentication*, volume 3072 of *LNCS*. Springer, 2004.

[305] Kentaro Toyama and Eric Horvitz. Bayesian modality fusion: Probabilistic integration of multiple vision algorithms for head tracking. In *Proc. Fourth Asian Conf. on Computer Vision (ACCV)*, Taipei, Taiwan, January 2000. held in conjunction with the 13th Nat. Conf. on Artificial Intelligence (AAAI-96).

[306] O. Tucha, D. Stasik, L. Mecklinger, I. Karl, and K.W. Lange. The effect of caffeine on handwriting movements in skilled writers. In *Proc. 12th Conf. of the International Graphonomics Society (IGS2005)*, 2005.

[307] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing. i. theory. *IEEE Trans. on Signal Processing*, 41(2):821–833, Feb. 1993. doi: 10.1109/78.193220.

[308] V. Vanhoucke and A. Sankar. Mixtures of inverse covariances. *IEEE Trans. Speech and Audio Processing*, 12(3):250–264, 2004. ISSN 1063-6676.

[309] V. Wan and S. Renals. Speaker verification using sequence discriminant support vector machines. *IEEE Transactions on Speech and Audio Processing*, 13(2):203–210, March 2005.

[310] Peiming Wang and Martin L. Puterman. Mixed logistic regression models. *Journal of Agricultural, Biological, and Environmental Statistics*, 3(2):175–200, June 1998.

[311] Larry Wasserman. *All of statistics - a concise course in statistical inference*. Springer, 2004.

[312] (James L.) Wayman, (Anil K.) Jain, Davide Maltoni, and Dario Maio. An introduction to biometric authentication systems. In (James L.) Wayman, (Anil K.) Jain, Davide Maltoni, and Dario Maio, editors, *Biometric Systems: Technology, Design and Performance Evaluation*, chapter 1, pages 1–20. Springer-Verlag, London, 2005.

[313] Klaus-D. Wernecke. A coupling procedure for the discrimination of mixed data. *Biometrics*, 48(2):497–506, June 1992.

[314] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufman, 2nd edition, 2005.

[315] S. K. Wong and C. Butz. Contextual weak independence in bayesian networks. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 670–67, San Francisco, CA, 1999. Morgan Kaufmann.

[316] K. Woods, Jr. Kegelmeyer, W.P., and K. Bowyer. Combination of multiple classifiers using local accuracy estimates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(4):405–410, 1997. ISSN 0162-8828.

[317] John D. Woodward, Christopher Horn, Julius Gatune, and Aryn Thomas. Biometrics: A look at facial recognition. Documented briefing, RAND, 2003.

[318] Nathaniel A. Woody and Steven D. Brown. Hybrid bayesian networks: making the hybrid bayesian classifier robust to missing training data. *Journal of Chemometrics*, 17(5):266–273, 2003.

[319] Q.-Z. Wu, I.-C. Jou, and S.-Y. Lee. On-line signature verification using LPC cepstrum and neural networks. *IEEE Trans. on systems, man and cybernetics, Part B*, 27(1):148–153, Feb. 1997.

[320] Yang Xiang. *Probabilistic Reasoning in Multiagent Systems: A Graphical Models Approach*. Cambridge University Press, 2002.

[321] Xuhong Xiao and Graham Leedham. Signature verification using a modified bayesian network. *Pattern Recognition*, 35(5):983–995, May 2002.

[322] L. Xu, A. Krzyzak, and C.Y. Suen. Methods of combining multiple classifiers and their applications tohandwriting recognition. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(3):418–435, 1992. ISSN 0018-9472.

[323] G. Xuan, W. Zhang, and P. Chai. EM algorithms of Gaussian Mixture Model and Hidden Markov Model. In *Proc. International Conference on Image Processing 2001*, pages 145–148, 2001.

[324] L. Yang, B.K. Widjaja, and R. Prasad. Application of hidden Markov models for signature verification. *Pattern Recognition*, 28(2):161–170, 1995.

[325] B. Yanikoglu and A. Kholmatov. An improved decision criterion for genuine/forgery classification in on-line signature verification. In *Proceedings ICANN/ICONIP 2003*, June 2003.

[326] Dit-Yan Yeung, Hong Chang, Yimin Xiong, Susan George, Ramanujan Kashi, Takashi Matsumoto, and Gerhard Rigoll. SVC2004: First international signature verification competition. In *Proceedings 2004 Biometric Authentication: First International Conference, (ICBA 2004)*, pages 16–22, Hong Kong, China, July 2004.

[327] H.S. Yoon, J.Y. Lee, and H.S. Yang. An on-line signature verification system using Hidden Markov Model in polar space. In *Proc. Eigth International Workshop on Frontiers in Handwriting Recognition*, pages 329–333, Aug. 2002.

[328] K. Yu, J. Mason, and J. Oglesby. Speaker recognition using hidden Markov models, dynamic time warping and vector quantisation. *IEE Proceedings - Vision, Image and Signal Processing*, 142:313–318, October 1995.

[329] Nevin Lianwen Zhang and David Poole. Exploiting causal independence in bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328, 1996.

[330] Nevin Lianwen Zhang and David Poole. On the role of context-specific independence in probabilistic inference. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 1288–1293, 1999.

[331] Nengheng Zheng, Tan Lee, and P. C. Ching. Integration of complementary acoustic features for speaker recognition. *Signal Processing Letters*, 14(3):181–184, 2007. ISSN 1070-9908.

[332] Rong Zheng, Shuwu Zhang, and Bo Xu. A comparative study of feature and score normalization for speaker verification. In *Proc. Int. Conf. on Biometrics (ICB)*, pages 531–538, Hong Kong, China, January 2006.

Appendix A

A.1 Benchmark Databases used

> It is a capital mistake to theorise before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts.
>
> Arthur Conan Doyle, *The adventures of Sherlock Holmes*

A.1.1 Signature databases

The MCYT database [218] contains signature and fingerprint data for 330 users. A 100-users subset of this database, called MCYT-100, is available to the members of the European BioSecure Network of Excellence. Each user provides 25 authentic signature samples (x, y, pressure, azimuth and elevation), and is forged 5 times by 5 different users, for a total 1000 authentic signatures and 1250 forgeries. The forgers are given time to practice on their target and are shown a static image of the target's signature.

The SVC 2004 database [326] is divided in two parts: an evaluation (training) set of 40 users which is freely available, and a sequestered set used in the competition, which is not distributed. Each user contributes 20 signatures, and is forged 20 times. The data is acquired in two sessions at least a week apart. The forgeries are performed by at least 4 different forgers, which are allowed to practice by watching a dynamic replay of the signing sequence. The data (for task 2) contains (x, y, pressure, azimuth, elevation, pen down status, time stamp) signals. A noteworthy information is that for privacy reasons, users were advised not to contribute their real signatures so this database contains alias signatures. This means the intra-user variability is probably overimportant. Also, this database contains both Chinese-style (ideograms) and latin-style (left-to right latin alphabet) signatures. Results are generally presented following the experimental protocol of the competition: all EERs are averaged over 10 crossvalidation run, during which 5 signatures out of the first 10 (first session) are randomly selected for training.

The BMEC2007 development database contains 50 users and is part of the larger BioSecure DS3

dataset. Signatures are acquired on a low-power mobile platform (Ipaq PDA). This means that some data is missing, and preprocessing approaches outlined in Section 4.5.2 have to be applied. Furthermore, the orientation of the signatures is haphazard. The acquisition platform only captures binary pressure (on/off) and x,y signals. No pen orientation information is available. The low quality of the data explains why error rates are in general higher on this database.

A.1.2 Speech databases

The BANCA database [14] contains speech data for 208 users, captured with 2 different microphones (one high-quality and one low-quality) in 12 sessions (three acoustical conditions). The data, about 40 seconds per session, consists of isolated digits and spontaneous speech, and is sampled at 32 kHz, quantised at 16 bits per sample, and recorded in mono. We use the english subset, consisting of 2x26 users. We generally follow the P protocol.

The XM2VTS database [192] contains 295 users and was recorded in 4 sessions about a month apart. It contains about 24 seconds of speech per user, read material (2 digits sequences and one sentence), for a total of total 7080 files. The files are sampled at 32 KHz and 16-bits quantised. While the amount of data per user is not very large and could lead to under-trained models, this database is one of the largest available for broadband speech. We generally follow the Lausanne protocol, configuration 1.

Since the signal quality in XM2VTS is high, we have also generated a noisy version of XM2VTS, by adding randomly-selected segments of babble-type noise recorded in a lively cafeteria environment, in SNRs uniformly distributed between 0 and 20 dB.

The CUAVE audio-visual database [223] is a labelled database containing 36 individual users, both male and female, each providing utterances of separated digits for about 2 minutes.

A.2 Curriculum vitae

Jonas Richiardi

rue Mauborget 3, 1003 Lausanne
P: +41/21/311 31 57
E: jonas.richiardi@epfl.ch

Nationality: Swiss and Italian
DoB: 14th of April 1976 (31 years old)

Work Experience

Apr. 2003 - present — EPFL, Signal Processing Institute
Lausanne, Switzerland
Research assistant: Worked on pattern recognition and signal processing for two European and 3 National projects. In addition to innovative research, duties include interacting with research teams Europe-wide, supervising an engineer for a software engineering project, and in-house software development.
Teaching assistant: Masters course on speech processing, practical laboratory on speech processing, masters course on biometrics. Duties include project management for master and term student projects.

1998-present — Intersource
Geneva, Switzerland
Senior trainer and consultant: Internet/Intranet development, consulting and training for private customers as well as companies (e.g. TetraPak, DuPont, Orange Communications), international organizations (e.g. UN, WHO, WIPO, IEC, ITU), and governments (city & county of Geneva, Kingdom of Morocco). Creation of Internet security courses for two large private banks in Geneva. Duties include advising clients on software and hardware architectures, and working with them to elicit specific functional requirements.

Education

Apr. 2003 - present — EPFL, Signal Processing Institute
Lausanne, Switzerland
Ph.D. student, working on signal processing and pattern recognition for multimodal biometrics, handwritten signature verification and speech-based authentication.

Oct. 2001 - Aug. 2002 — University of Cambridge (Darwin College), Computer Laboratory/Engineering Department
Cambridge, United Kingdom
MPhil Computer Speech, Text and Internet Technology.

Oct. 1999 - Jun. 2001 — University of Essex, Department of Electronic Systems Engineering

Colchester, United Kingdom
BEng (Hons.) Electronic Engineering, 1st class honours.

Oct. 1998- Jul. 1999 University of Derby, School of Engineering

Derby, United Kingdom
First year of BSc (Hons.) Music Technology & Audio Systems Design completed, yearly average: A. Transfer to University of Essex with direct level 3 entry.

Languages

French
Mother tongue.

English
Excellent command; degree and postgraduate studies as well as teaching experience in English.

Italian
Very good command; Italian citizenship.

German
Good command; 8 years of study; GfDS' "Deutsch als Fremdsprache" Mittelstufe II diploma ("good" overall mark); work experience in German.

Spanish
Beginner level; basic conversation abilities.

Japanese
Some basics; 2 years of study; simple conversation abilities, elementary reading skills.

System design skills

Software design
C/C++, Matlab, Python: good knowledge, development on Linux and Windows platforms.
Java, PERL, Visual Basic/VB.NET: basic knowledge.
Web/Intranet programming: XHTML, CSS, JavaScript, XML&XSL, PHP, ASP: training and consulting for all levels.

Electronics Design
By training, good digital design skills and good knowledge of Verilog HDL. Basic analogue & VLSI design, test and measurement skills.

Signal processing and Pattern recognition
Good working knowledge of analogue and digital signal processing, strong interest for speech processing and recognition, pattern classification, and machine learning. Use of Matlab, Torch, BNT, Alize, and HTK toolkits for pattern classification applications

Awards

2005	Institute of Electrical and Electronics Engineers best student paper award at the ICASSP 2005 conference.
	International Society of Motor Control best graduate student presentation award at the IGS 2005 conference.
	European Association for Signal, Speech and Image Processing student paper competition finalist at the EUSIPCO 2005 conference.
2001	Cambridge Overseas Trust scholarship.
	Institution of Electrical Engineers Prize (best performance of the year across all BEng students at Essex University)
	British Telecom/BTExact Project Prize (amongst 3 best projects of the year at Essex University)

A.3 List of Publications

A.3.1 Journal papers

1. [167] Krzysztof Kryszczuk, Jonas Richiardi, Plamen Prodanov, and Andrzej Drygajlo. Reliability-based decision fusion in multimodal biometric verification systems. *EURASIP Journal of Advances in Signal Processing*, 2007. (in press).

2. [237] Plamen Prodanov, Andrzej Drygajlo, Jonas Richiardi, and Anil Alexander. Grounding in multimodal service robot conversational system using graphical models. *Journal of Intelligent Service Robotics*, 2007.

3. [69] Damien Dessimoz, Jonas Richiardi, Christophe Champod, and Andrzej Drygajlo. Multimodal biometrics for identity documents (MBioID). *Forensic Science International*, 167: 154–159, April 2007.

4. [260] Jonas Richiardi, Andrzej Drygajlo, and Plamen Prodanov. Confidence and reliability measures in speaker verification. *Journal of the Franklin Institute*, 343(6):574–595, September 2006.

A.3.2 Conference Papers

1. [258] Jonas Richiardi and Andrzej Drygajlo. Evaluation of speech quality measures for the purpose of speaker verification. In *Proc. Odyssey 2008: The Speaker and Language Recognition Workshop*, Stellenbosch, South Africa, January 2008.

2. [262] Jonas Richiardi, Krzysztof Kryszczuk, and Andrzej Drygajlo. Quality measures in unimodal and multimodal biometric verification. In *Proc. 15th European Signal Processing Conf. (EUSIPCO)*, Poznan, Poland, 2007.

3. [257] Jonas Richiardi and Andrzej Drygajlo. Reliability-based voting schemes using modality-independent features in multi-classifier biometric authentication. In *Proc. 7th Int. Workshop on Multiple Classifier Systems*, Prague, Czech Republic, May 2007. Springer.

4. [166] Krzysztof Kryszczuk, Jonas Richiardi, and Andrzej Drygajlo. Reliability estimation for multimodal error prediction and fusion. In *Proc. 7th Int. Workshop on Pattern Recognition in Information Systems (PRIS 2007)*, Funchal, Portugal, June 2007.

5. [153] J Kittler, N Poh, O Fatukasi, K Messer, K Kryszczuk, J Richiardi, and A Drygajlo. Quality dependent fusion of intramodal and multimodal biometric experts. In *Proc. SPIE Defense and Security Symposium*, Orlando, USA, April 2007.

6. [261] Jonas Richiardi, Plamen Prodanov, and Andrzej Drygajlo. Speaker verification with confidence and reliability measures. In *Proc. 2006 IEEE International Conference on Speech, Acoustics and Signal Processing*, Toulouse, France, May 2006.

7. [183] Marcus Liwicki, Andreas Schlapbach, Horst Bunke, Samy Bengio, Johnny Mariéthoz, and Jonas Richiardi. Writer identification for smart meeting room systems. In *Proc. 7th IAPR International Workshop on Document Analysis Systems*, volume 3872 of *Lecture Notes in Computer Science*, pages 186–195, Nelson, New Zealand, February 2006.

8. [253] J. Richiardi, A. Drygajlo, A. Palacios-Venin, R. Ludvig, O. Genton, and L. Houmgny. A distributed multimodal biometric authentication framework. In *Proc. 3nd COST 275 Workshop on Biometrics on the Internet*, pages 85–88, Hatfield, U.K., October 2005.

9. [165] Krzysztof Kryszczuk, Jonas Richiardi, Plamen Prodanov, and Andrzej Drygajlo. Error handling in multimodal biometric systems using reliability measures. In *Proc. 12th European Conference on Signal Processing (EUSIPCO)*, Antalya, Turkey, September 2005.

10. [259] Jonas Richiardi, Hamed Ketabdar, and Andrzej Drygajlo. Local and global feature selection for on-line signature verification. In *Proc. IAPR 8th International Conference on Document Analysis and Recognition (ICDAR 2005)*, volume 2, pages 625–629, Seoul, Korea, August-September 2005.

11. [146] H. Ketabdar, J. Richiardi, and A. Drygajlo. Global feature selection for on-line signature verification. In *Proc. 12th Conference of the International Graphonomics Society*, pages 59–63, Salerno, Italy, June 2005.

12. [236] P. Prodanov, J. Richiardi, and A. Drygajlo. Graphical models for dialogue repair in multimodal interaction with service robots. In *Proc. 8th COST276 workshop*, Trondheim, Norway, May 2005.

13. [254] J. Richiardi, P. Prodanov, and A. Drygajlo. A probabilistic measure of modality reliability in speaker verification. In *Proc. IEEE International Conf. on Acoustics, Speech and Signal Processing 2005*, pages 709–712, Philadelphia, USA, March 2005.

14. [252] J. Richiardi, J. Fierrez-Aguilar, J. Ortega-Garcia, and A. Drygajlo. On-line signature verification resilience to packet loss in IP networks. In *Proc. 2nd COST 275 Workshop on Biometrics on the Internet: fundamentals, advances and applications*, pages 9–14, Vigo, Spain, March 2004.

15. [251] Gaussian mixture models for on-line signature verification. In *Proc. ACM SIGMM Multimedia, Workshop on Biometrics methods and applications (WBMA)*, pages 115–122, Berkeley, USA, Nov. 2003.

A.3.3 Research reports

1. [255] Jonas Richiardi and Andrzej Drygajlo. Applying biometrics to identity documents: Estimating and coping with errors. SNSF project technical report, Swiss Federal Institute of Technology, October 2006.

2. [256] Jonas Richiardi and Andrzej Drygajlo. Applying biometrics to identity documents: Implementation issues. SNSF project technical report, Swiss Federal Institute of Technology, 2006.

3. [68] Damien Dessimoz, Jonas Richiardi, Christophe Champod, and Andrzej Drygajlo. Multimodal biometrics for identity documents: State-of-the-art. Technical Report PFS 341-08.05, University of Lausanne and EPFL, September 2005.

4. [250] J. Richiardi. Resilience of on-line signature verification to packet loss on IP networks: preliminary experiments. COST275 STSM Report, Fondazione Ugo Bodoni, Italy, Sept. 2003.